To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

User’s Manual

μPD78F0714

8-Bit Single-Chip Microcontroller

μPD78F0714

Document No. U16928EJ2V0UD00 (2nd edition)
Date Published September 2007 NS

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

② HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

⑤ POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current.

The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.

Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

EEPROM is a trademark of NEC Electronics Corporation.

Windows, Windows NT, and Windows XP are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

PC/AT is a trademark of International Business Machines Corporation.

HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.

SPARCstation is a trademark of SPARC International, Inc.

Solaris and SunOS are trademarks of Sun Microsystems, Inc.

TRON is an abbreviation of The Realtime Operating system Nucleus.

ITRON is an abbreviation of Industrial TRON.

SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, inc.

- **The information in this document is current as of August, 2007. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

(Note)

(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

INTRODUCTION

Readers

This manual is intended for user engineers who wish to understand the functions of the μPD78F0714 and design and develop application systems and programs for this device.
The target product is as follows.

μPD78F0714

Purpose

This manual is intended to give users an understanding of the functions described in the **Organization** below.

Organization

The μPD78F0714 manual is separated into two parts: this manual and the instructions edition (common to the 78K/0 Series).

μPD78F0714 User's Manual (This Manual)	**78K/0 Series User's Manual Instructions**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.
- How to interpret the register format:
  → For a bit number enclosed in brackets, the bit name is defined as a reserved word in the assembler, and is already defined in the header file named sfrbit.h in the C compiler.
- To check the details of a register when you know the register name.
  → See **APPENDIX B REGISTER INDEX**.
- To know details of the 78K/0 Series instructions.
  → Refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

Conventions

Data significance:	Higher digits on the left and lower digits on the right	
Active low representations:	$\overline{\times\times\times}$ (overscore over pin and signal name)	
Note:	Footnote for item marked with **Note** in the text.	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numerical representations:	Binary	···×××× or ××××B
	Decimal	···××××
	Hexadecimal	···××××H

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Documents Related to Devices

Document Name	Document No.
μPD78F0714 User's Manual	This manual
78K/0 Series Instructions User's Manual	U12326E

<R>

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
RA78K0 Ver. 3.80 Assembler Package	Operation	U17199E
	Language	U17198E
	Structured Assembly Language	U17197E
CC78K0 Ver. 3.70 C Compiler	Operation	U17201E
	Language	U17200E
ID78K0-QB Ver. 2.94 Integrated Debugger	Operation	U18330E
PM+ Ver. 5.20		U16934E

<R>

Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
QB-780714 In-Circuit Emulator	U17081E
QB-78K0MINI On-Chip Debug Emulator	U17029E
QB-78K0MINI2 On-Chip Debug Emulator with Programming Function	U18371E

Documents Related to Flash Memory Programming

Document Name	Document No.
PG-FP4 Flash Memory Programmer User's Manual	U15260E

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the "Semiconductor Device Mount Manual" website (http://www.necel.com/pkg/en/mount/index.html).

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

○ Minimum instruction execution time can be changed from high speed (0.1 μs: @ 20 MHz operation with X1 input clock) to low-speed (8.33 μs: @ 240 kHz operation with internal oscillation clock)

○ General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)

○ On-chip multiplier/divider

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16 bits remainder (division)

○ ROM, RAM capacities

Item / Part Number	Program Memory (ROM)		Data Memory (Internal High-Speed RAM)
μPD78F0714	Flash memory	32 KB	1024 bytes

○ On-chip single-power-supply flash memory

○ Self-programming (with boot swap function)

○ On-chip debug function

○ On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)

○ Short startup is possible via the CPU default start using the internal oscillator

○ On-chip watchdog timer (operable with internal oscillation clock)

○ On-chip clock output/buzzer output controller00

○ On-chip real-time output ports

○ I/O ports: 48

○ Timer: 7 channels

○ Serial interface: 2 channels (UART: 1 channel, CSI: 1 channel)

○ 10-bit resolution A/D converter: 8 channels

○ Supply voltage: V_{DD} = 4.0 to 5.5 V

○ Operating ambient temperature: T_A = –40 to +85°C

<R>

1.2 Applications

○ Household electrical appliances
- Refrigerator
- Dish washer
- Washing machine, Dryer
- Outdoor air conditioner units
- Microwave ovens, electric rice cookers

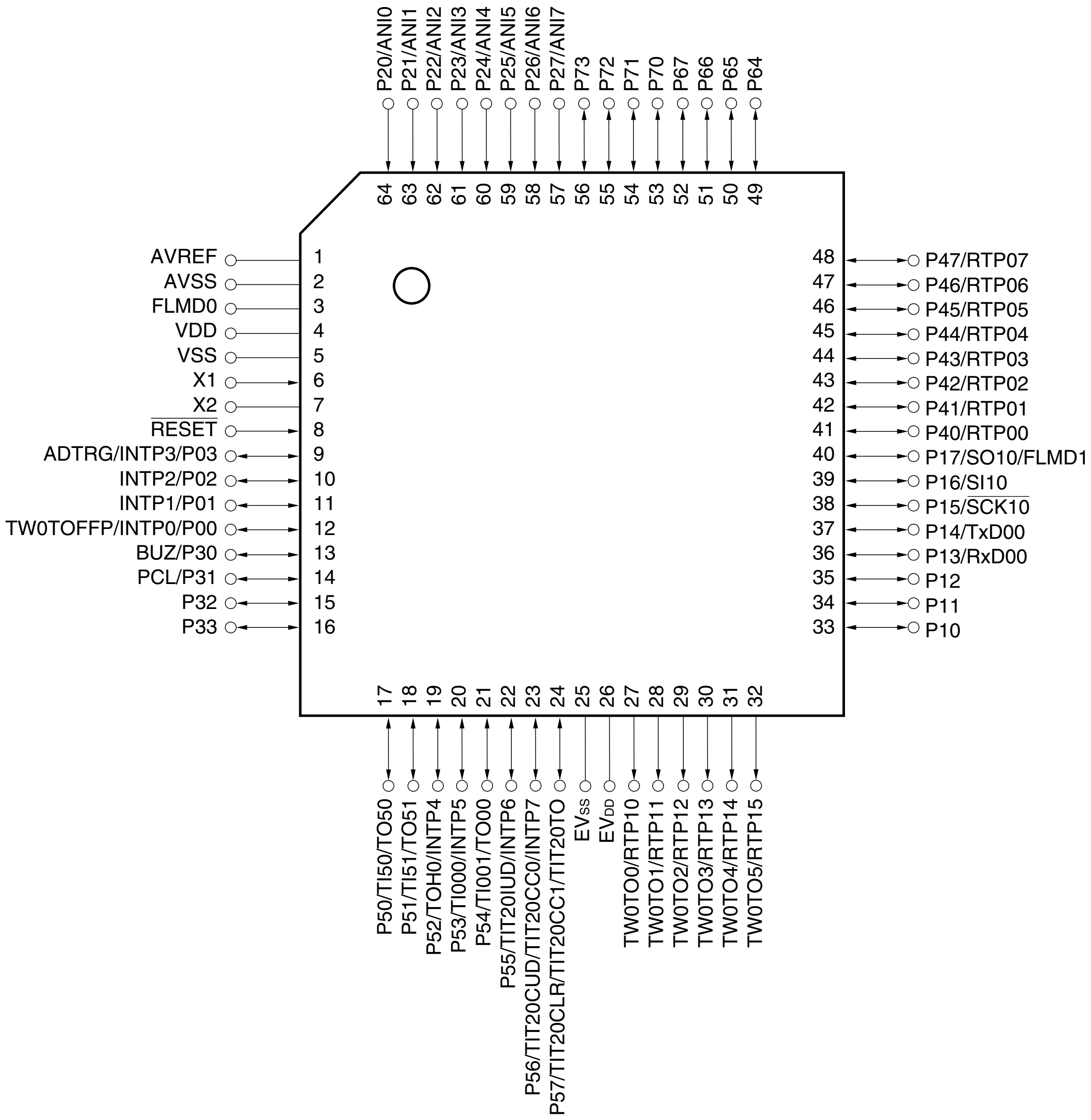
○ Industrial equipment
- Pumps

1.3 Ordering Information

Part Number	Package
μPD78F0714GK-9ET	64-pin plastic TQFP (fine pitch) (12 × 12)

1.4 Pin Configuration (Top View)

- 64-pin plastic TQFP (fine pitch) (12 × 12)

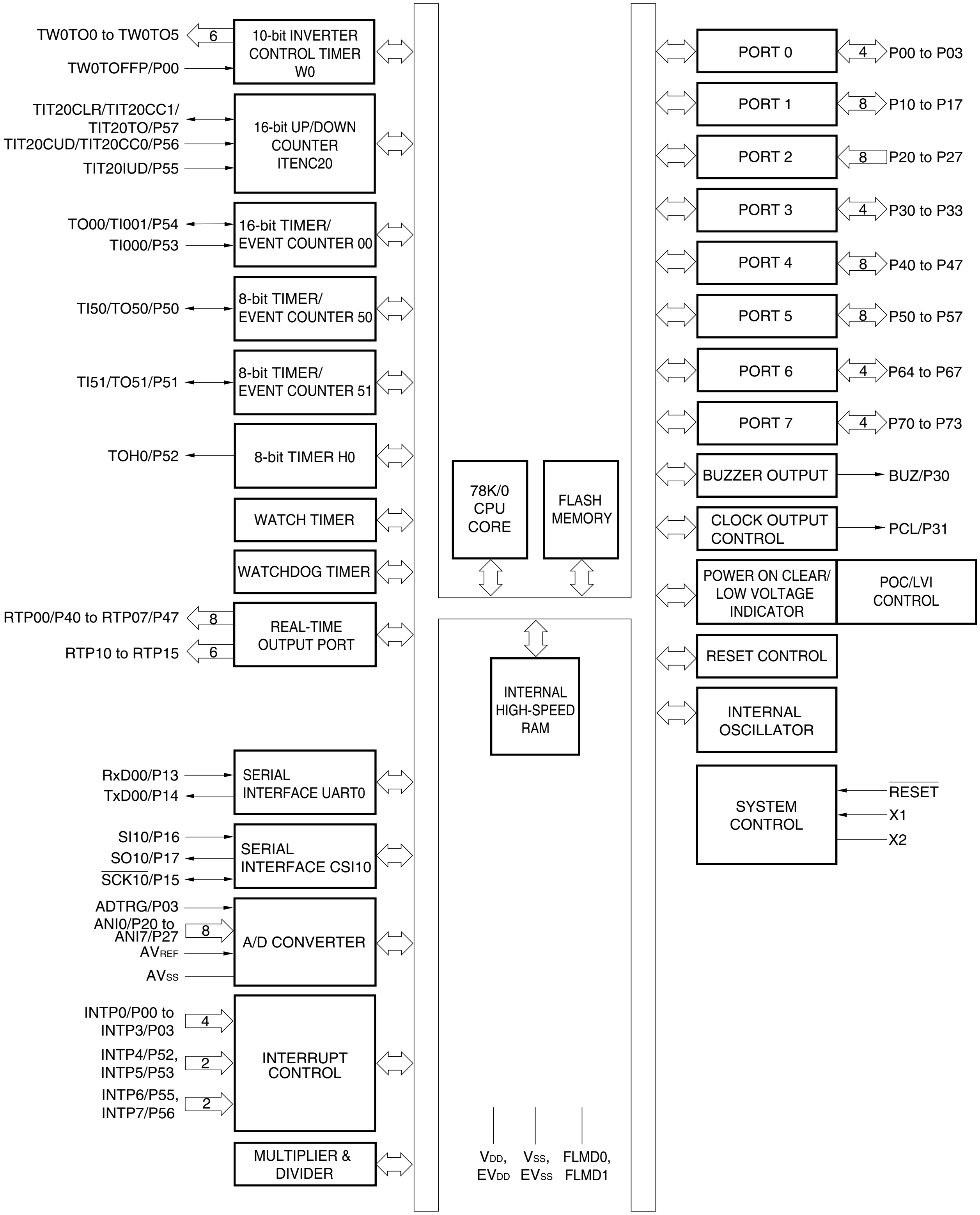


Caution Connect the AV$_{SS}$ pin to V$_{SS}$.

Pin Identification

ADTRG:	A/D trigger input
ANI0 to ANI7:	Analog input
AV$_{\text{REF}}$:	Analog reference voltage
AV$_{\text{SS}}$:	Analog ground
BUZ:	Buzzer output
EV$_{\text{DD}}$:	Power supply for port
EV$_{\text{SS}}$:	Ground for port
FLMD0, FLMD1:	Flash programming mode
INTP0 to INTP7:	External interrupt input
P00 to P03:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P33:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P64 to P67:	Port 6
P70 to P73:	Port 7
PCL:	Programmable clock output
$\overline{\text{RESET}}$:	Reset
RTP00 to RTP07:	Real-time output port
RTP10 to RTP15:	Real-time output port
RxD00:	Receive data
$\overline{\text{SCK10}}$	Serial clock input/output
SI10:	Serial data input
SO10:	Serial data output
TI000, TI001:	Timer input
TI50, TI51:	Timer input
TIT20CLR:	Up/down counter clear
TIT20CUD:	Up/down counter clock select
TIT20CC0, TIT20CC1:	Up/down counter capture input
TIT20IUD:	Up/down counter clock
TIT20TO:	Up/down counter output
TO00:	Timer output
TO50, TO51:	Timer output
TOH0:	Timer output
TW0TO0 to TW0TO5:	Timer output
TW0TOFFP:	Timer output off
TxD00:	Transmit data
V$_{\text{DD}}$:	Power supply
V$_{\text{SS}}$:	Ground
X1, X2:	Crystal oscillator (X1 input clock)

1.5 Block Diagram



1.6 Outline of Functions

Item		μPD78F0714
Internal memory	Flash memory (self-programming supported)	32 KB
	High-speed RAM	1 KB
Memory space		64 KB
X1 input clock (oscillation frequency)		Ceramic/crystal/external clock oscillation [20 MHz (V_{DD} = 4.0 to 5.5 V)]
Internal oscillation clock (oscillation frequency)		Internal oscillator (240 kHz (TYP.))
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)
Minimum instruction execution time		0.1 μs/0.2 μs/0.4 μs/0.8 μs/1.6 μs (X1 input clock: @ f_{XP} = 20 MHz operation)
		8.3 μs/16.6 μs/33.2 μs/66.4 μs/132.8 μs (TYP.) (Internal oscillation clock: @ f_R = 240 kHz (TYP.) operation)
Instruction set		• 16-bit operation • Multiply/divide (8 bits × 8 bits, 16 bits ÷ 8 bits) • Bit manipulate (set, reset, test, and Boolean operation) • BCD adjust, etc.
I/O ports		Total: 48 CMOS I/O 40 CMOS input 8
Timers		• 10-bit inverter control timer: 1 channel • 16-bit up/down counter: 1 channel • 16-bit timer/event counter: 1 channel • 8-bit timer/event counter: 2 channels • 8-bit timer: 1 channel • Watchdog timer: 1 channel
	Timer outputs	11 (inverter control output: 6)
Clock output		156.25 kHz, 312.5 kHz, 625 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz, 20MHz (X1 input clock: 20 MHz)
Buzzer output		2.44 kHz, 4.88 kHz, 9.77 kHz, 19.5kHz (X1 input clock: 20 MHz)
Real-time output ports		• 8 bits × 1 or 4 bits × 2 • 6 bits × 1 or 4 bits × 2
A/D converter		10-bit resolution × 8 channels
Serial interface		• UART mode: 1 channel • 3-wire serial I/O mode: 1 channel
Multiplier/divider		• 16 bits × 16 bits = 32 bits (multiplication) • 32 bits ÷ 16 bits = 32 bits remainder of 16 bits (division)
<R> Vectored interrupt sources	Internal	Non-maskable: 1, Maskable: 19
	External	8
Reset		• Reset using $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector
Supply voltage		V_{DD} = 4.0 to 5.5 V
Operating ambient temperature		T_A = –40 to +85°C
Package		64-pin plastic TQFP (fine pitch) (12 × 12)

An outline of the timer is shown below.

		10-Bit Inverter Control Timer	16-Bit Up/down Counter ITENC20	16-Bit Timer/ Event Counter 00	8-Bit Timer/ Event Counters 50 and 51		8-Bit Timer H0	Watchdog Timer
					TM50	TM51		
Operation mode	Interval timer	1 channel	1 channel	1 channel	1 channel	1 channel	1 channel	–
	External event counter	–	1 channel	1 channel	1 channel	1 channel	–	–
Function	Timer output	6 outputs	1 output	1 output	1 output	1 output	1 output	–
	PPG output	–	–	1 output	–	–	–	–
	PWM output	6 outputs	1 output	–	1 output	1 output	1 output	–
	Pulse width measurement	–	–	2 inputs	–	–	–	–
	Square-wave output	–	1 output	1 output	1 output	1 output	1 output	–
	Watchdog Timer	–	–	–	–	–	–	1 channel
	Interrupt source	4	4	2	1	1	1	–

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are three types of pin I/O buffer power supplies: AV_{REF}, EV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27
EV_{DD}	Port pins other than P20 to P27
V_{DD}	Pins other than port pins

(1) Port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0/TW0TOFFP
P01				INTP1
P02				INTP2
P03				INTP3/ADTRG
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–
P11				–
P12				–
P13				RxD00
P14				TxD00
P15				$\overline{\text{SCK10}}$
P16				SI10
P17				SO10/FLMD1
P20 to P27	Input	Port 2. 8-bit input-only port.	Input	ANI0 to ANI7
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	BUZ
P31				PCL
P32				–
P33				–
P40 to P47	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	RTP00 to RTP07

(1) Port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI50/TO50
P51				TI51/TO51
P52				TOH0/INTP4
P53				TI000/INTP5
P54				TI001/TO00
P55				TIT20IUD/INTP6
P56				TIT20CUD /TIT20CC0/INTP7
P57				TIT20CLR /TIT20CC1 /TIT20TO
P64 to P67	I/O	Port 6. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–
P70 to P73	I/O	Port 7. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–

(2) Non-port pins (1/2)

Pin Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input	P00/TW0TOFFP
INTP1				P01
INTP2				P02
INTP3				P03/ADTRG
INTP4				P52/TOH0
INTP5				P53/TI000
INTP6				P55/TIT20IUD
INTP7				P56/TIT20CC0 /TIT20CUD
SI10	Input	Serial data input to serial interface	Input	P16
SO10	Output	Serial data output from serial interface	Input	P17/FLMD1
$\overline{\text{SCK10}}$	I/O	Clock input/output for serial interface	Input	P15
RxD00	Input	Serial data input to asynchronous serial interface	Input	P13
TxD00	Output	Serial data output from asynchronous serial interface	Input	P14
TW0TOFFP	Input	External input to stop 10-bit inverter control timer output	Input	P00/INTP0
TW0TO0- TW0TO5	Output	10-bit inverter control timer output	Output	RTP10-RTP15
TIT20IUD	Input	External count clock input to 16-bit up/down counter	Input	P55/INTP6
TIT20CUD		Count operation switching input to 16-bit up/down counter		P56/TIT20CC0 /INTP7
TIT20CC0		External capture trigger input to 16-bit up/down counter		P56/TIT20CUD /INTP7
TIT20CC1				P57/TIT20CLR /TIT20TO
TIT20CLR		External clear input to 16-bit up/down counter		P57/TIT20CC1 /TIT20TO
TIT20TO	Output	Pulse signal output of 16-bit up/down counter	Input	P57/TIT20CLR /TIT20CC1
TI000	Input	External count clock input to 16-bit timer/event counter 00 Capture trigger input to capture registers (CR000, CR010) of 16-bit timer/event counter 00	Input	P53/INTP5
TI001		Capture trigger input to capture register (CR000) of 16-bit timer/event counter 00		P54/TO00
TO00	Output	16-bit timer/event counter 00 output	Input	P54/TI001
TI50	Input	External count clock input to 8-bit timer/event counter 50	Input	P50/TO50
TI51		External count clock input to 8-bit timer/event counter 51		P51/TO51
TO50	Output	8-bit timer/event counter 50 output	Input	P50/TI50
TO51		8-bit timer/event counter 51 output		P51/TI51
TOH0		8-bit timer H0 output		P52/INTP4

(2) Non-port pins (2/2)

Pin Name	I/O	Function	After Reset	Alternate Function
PCL	Output	Clock output (for trimming of X1 input clock)	Input	P31
BUZ	Output	Buzzer output	Input	P30
RTP00 to RTP07	Output	Real-time output port 0 output	Input	P40 to P47
RTP10 to RTP15	Output	Real-time output port 1 output	Output	TW0TO0 to TW0TO5
ADTRG	Input	A/D converter trigger input	Input	P03/INTP3
ANI0 to ANI7	Input	A/D converter analog input	Input	P20 to P27
AV_{REF}	Input	A/D converter reference voltage input and positive power supply for port 2	–	–
AV_{SS}	–	A/D converter ground potential. Make the same potential as EV_{SS} or V_{SS}.	–	–
$\overline{RESET}$	Input	System reset input	–	–
X1	Input	Connecting resonator for X1 input clock oscillation	–	–
X2	–		–	–
V_{DD}	–	Positive power supply (except for ports)	–	–
EV_{DD}	–	Positive power supply for ports	–	–
V_{SS}	–	Ground potential (except for ports)	–	–
EV_{SS}	–	Ground potential for ports	–	–
FLMD0	–	Flash memory programming mode setting	–	–
FLMD1	Input		Input	P17/SO10

2.2 Description of Pin Functions

2.2.1 P00 to P03 (port 0)

P00 to P03 function as a 4-bit I/O port. These pins also function as external interrupt request input, timer output stop external signal, and A/D converter trigger input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P00 to P03 function as a 4-bit I/O port. P00 to P03 can be set to input or output in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

(2) Control mode

P00 to P03 function as external interrupt request input, timer output stop external signal, and A/D converter trigger input.

(a) INTP0 to INTP3

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified. INTP2 also functions as an external trigger signal input pin of the real-time output port when a valid edge is input.

(b) TW0TOFFP

This is an external input pin to stop timer output (TW0TO0 to TW0TO5).

(c) ADTRG

This is an external trigger signal input pin of the A/D converter.

2.2.2 P10 to P17 (port 1)

P10 to P17 function as an 8-bit I/O port. These pins also function as pins for serial interface data I/O, clock I/O, and flash memory programming mode setting.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P10 to P17 function as an 8-bit I/O port. P10 to P17 can be set to input or output in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P17 function as serial interface data I/O and clock I/O.

(a) SI10

This is a serial interface serial data input pin.

(b) SO10

This is a serial interface serial data output pin.

(c) $\overline{\textbf{SCK10}}$

This is a serial interface serial clock I/O pin.

(d) RxD00

This is the serial data input pin of the asynchronous serial interface.

(e) TxD00

This is the serial data output pin of the asynchronous serial interface.

(f) FLMD1

This pin sets the flash memory programming mode.

2.2.3 P20 to P27 (port 2)

P20 to P27 function as an 8-bit input-only port. These pins also function as pins for A/D converter analog input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P20 to P27 function as an 8-bit input-only port.

<R> **Caution Use P20 to P27 at EV_{DD} = AV_{REF} when using them in the port mode.**

(2) Control mode

P20 to P27 function as A/D converter analog input pins (ANI0 to ANI7). When using these pins as analog input pins, see **(5) ANI0/P20 to ANI7/P27** in **15.6 Cautions for A/D Converter**.

2.2.4 P30 to P33 (port 3)

P30 to P33 function as a 4-bit I/O port. These pins also function as pins for clock output, and buzzer output.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P30 to P33 function as a 4-bit I/O port. P30 to P33 can be set to input or output in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P33 function as clock output, and buzzer output pins.

(a) BUZ

This is a buzzer output pin.

(b) PCL

This is a clock output pin.

Caution Be sure to pull down P31 after reset to prevent malfunction.

Remark The P31 and P32 pins of the μPD78F0714 can be used to set the on-chip debug mode when the on-chip debug function is used. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

2.2.5 P40 to P47 (port 4)

P40 to P47 function as an 8-bit I/O port. These pins also function as real-time output port pins.

The following operation modes can be specified.

(1) Port mode

P40 to P47 function as an 8-bit I/O port. P40 to P47 can be set to input or output in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

(2) Control mode

P40 to P47 function as the pins for the real-time output port (RTP00 to RTP07) that outputs data in synchronization with a trigger.

2.2.6 P50 to P57 (port 5)

P50 to P57 function as an 8-bit I/O port. These pins also function as external interrupt request input and timer I/O.

The following operation modes can be specified.

(1) Port mode

P50 to P57 function as an 8-bit I/O port. P50 to P57 can be set to input or output in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 to P57 function as the pins for the external interrupt request input and timer I/O.

(a) INTP4 to INTP7

These are the external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

(b) TI50, TI51

These are the pins for inputting an external count clock to 8-bit timer/event counter 50 and 51.

(c) TO50, TO51

These are timer output pins from 8-bit timer/event counters 50 and 51.

(d) TI000

This is the pin for inputting an external count clock to 16-bit timer/event counters 00 and is also for inputting a capture trigger signal to the capture registers (CR00, CR01).

(e) TI001

This is the pin for inputting a capture trigger signal to the capture register (CR00) of 16-bit timer/event counters 00.

(f) TO00, TOH0

These are timer output pins from 16-bit timer/event counter 00 and 8-bit timer H0.

(g) TIT20IUD

This is the pin for inputting an external count clock to 16-bit up/down counter ITENC20.

(h) TIT20IUD

This is the pin for inputting an count operation switching signal to 16-bit up/down counter ITENC20.

(i) TIT20CLR

This is the pin for inputting a clear signal to 16-bit up/down counter ITENC20.

(j) TIT20CC0, TIT20CC1

These are the pins for inputting an external capture trigger to 16-bit up/down counter ITENC20.

(k) TIT20TO

This is a 16-bit up/down counter ITENC20 output pin.

2.2.7 P64 to P67 (port 6)

P64 to P67 function as a 4-bit I/O port. P64 to P67 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

Use of an on-chip pull-up resistor can be specified for P64 to P67 by pull-up resistor option register 6 (PU6).

2.2.8 P70 to P73 (port 7)

P70 to P73 function as a 4-bit I/O port. P70 to P73 can be set to input or output in 1-bit units using port mode register 7 (PM7). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

2.2.9 TW0TO0/RTP10 to TW0TO5/RTP15

These are 10-bit inverter control timer output pins.

And, these pins function also as real-time output port pins.

2.2.10 AV_{REF}

This is the A/D converter reference voltage input pin.

When the A/D converter is not used, connect this pin directly to EV_{DD} or V_{DD}[Note].

Note Connect port 2 directly to EV_{DD} when it is used as a digital port.

2.2.11 AV_{SS}

This is the A/D converter ground potential pin. Even when the A/D converter is not used, always use this pin with the same potential as the EV_{SS} pin or V_{SS} pin.

2.2.12 $\overline{\text{RESET}}$

This is the active-low system reset input pin.

2.2.13 X1 and X2

These are the pins for connecting a resonator for the X1 input clock.

When supplying an external clock, input a signal to the X1 pin and input the inverse signal to the X2 pin.

Remark The X1 and X2 pins of the product with an on-chip debug function (part number pending) can be used to set the on-chip debug mode when the on-chip debug function is used. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

2.2.14 V_{DD} and EV_{DD}

V_{DD} is the positive power supply pin for other than ports.

EV_{DD} is the positive power supply pin for ports.

2.2.15 V_{SS} and EV_{SS}

V_{SS} is the ground potential pin for other than ports.

EV_{SS} is the ground potential pin for ports.

2.2.16 FLMD0

This pin sets the flash memory programming mode.

Connect FLMD0 to a flash memory programmer in the flash memory programming mode, and to EV_{SS} or V_{SS} in the normal operation mode.

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

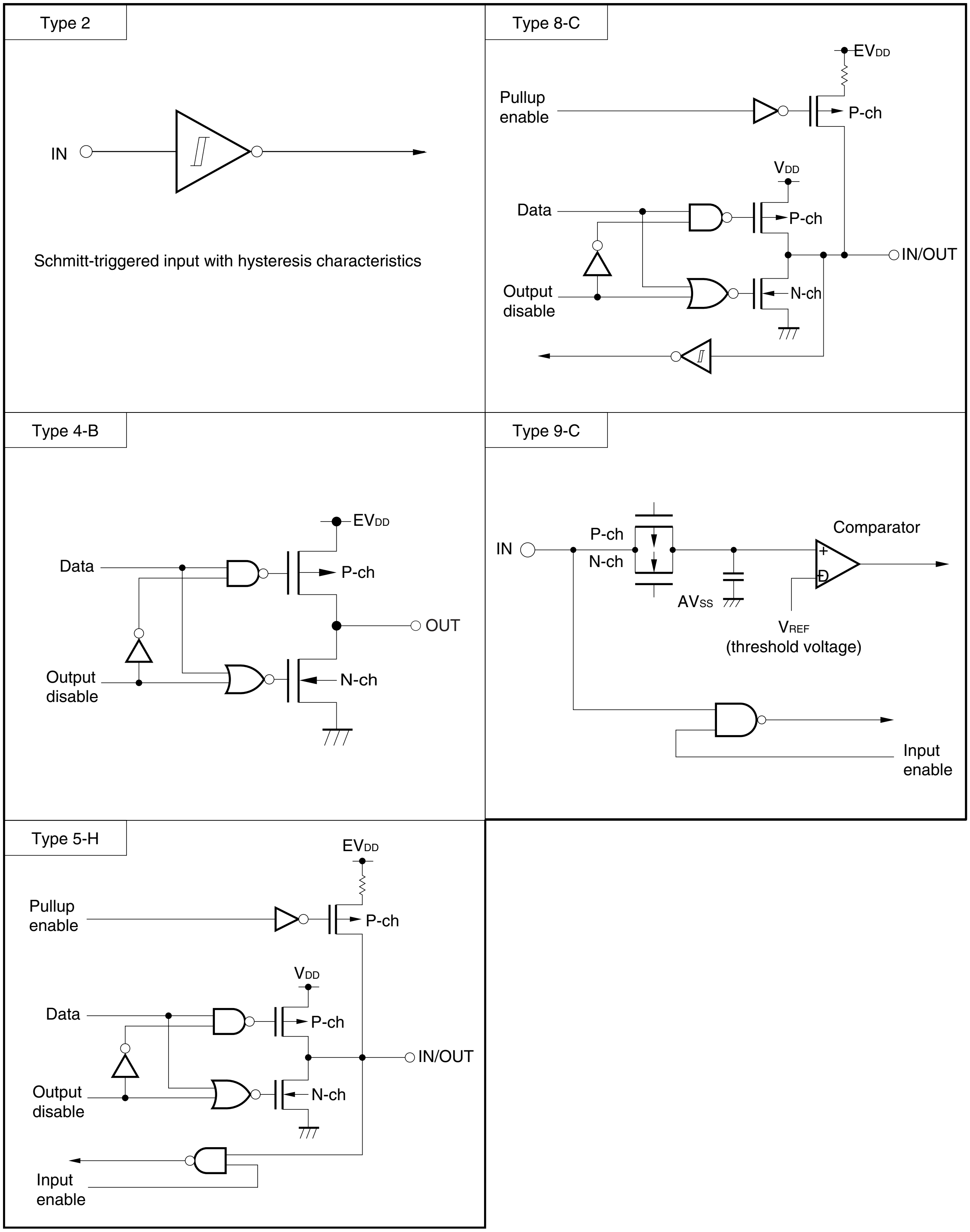
Table 2-2 shows the types of pin I/O circuits and the recommended connections of unused pins.
See **Figure 2-1** for the configuration of the I/O circuit of each type.

Table 2-2. Pin I/O Circuit Types

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00/INTP0/TW0TOFFP	8-C	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01/INTP1			
P02/INTP2			
P03/INTP3/ADTRG			
P10			
P11			
P12			
P13/RxD00			
P14/TxD00	5-H		
P15/ $\overline{\text{SCK10}}$	8-C		
P16/SI10			
P17/SO10/FLMD1	5-H		
P20/ANI0 to P27/ANI7	9-C	Input	Connect to EV_{DD} or EV_{SS}.
P30/BUZ	5-H	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31/PCL			
P32			
P33			
P40/RTP00 to P47/RTP07			
P50/TI50/TO50	8-C		
P51/TI51/TO51			
P52/TOH0/INTP4			
P53/TI000/INTP5			
P54/TI001/TO00			
P55/TIT20IUD/INTP6			
P56/TIT20CUD/TIT20CC0/INTP7			
P57/TIT20CLR/TIT20CC1/TIT20TO			
P64 to P67	5-H		
P70 to P73			
TW0TO0/RTP10-TW0TO5/RTP15	4-B	Output	Leave open.
$\overline{\text{RESET}}$	2	Input	–
AV_{REF}	–	–	Connect directly to EV_{DD} or V_{DD}[Note].
AV_{SS}			Connect directly to EV_{SS} or V_{SS}.
FLMD0			Connect to EV_{SS} or V_{SS}.

Note Connect port 2 directly to EV_{DD} when it is used as a digital port.

Figure 2-1. Pin I/O Circuit List



Schmitt-triggered input with hysteresis characteristics

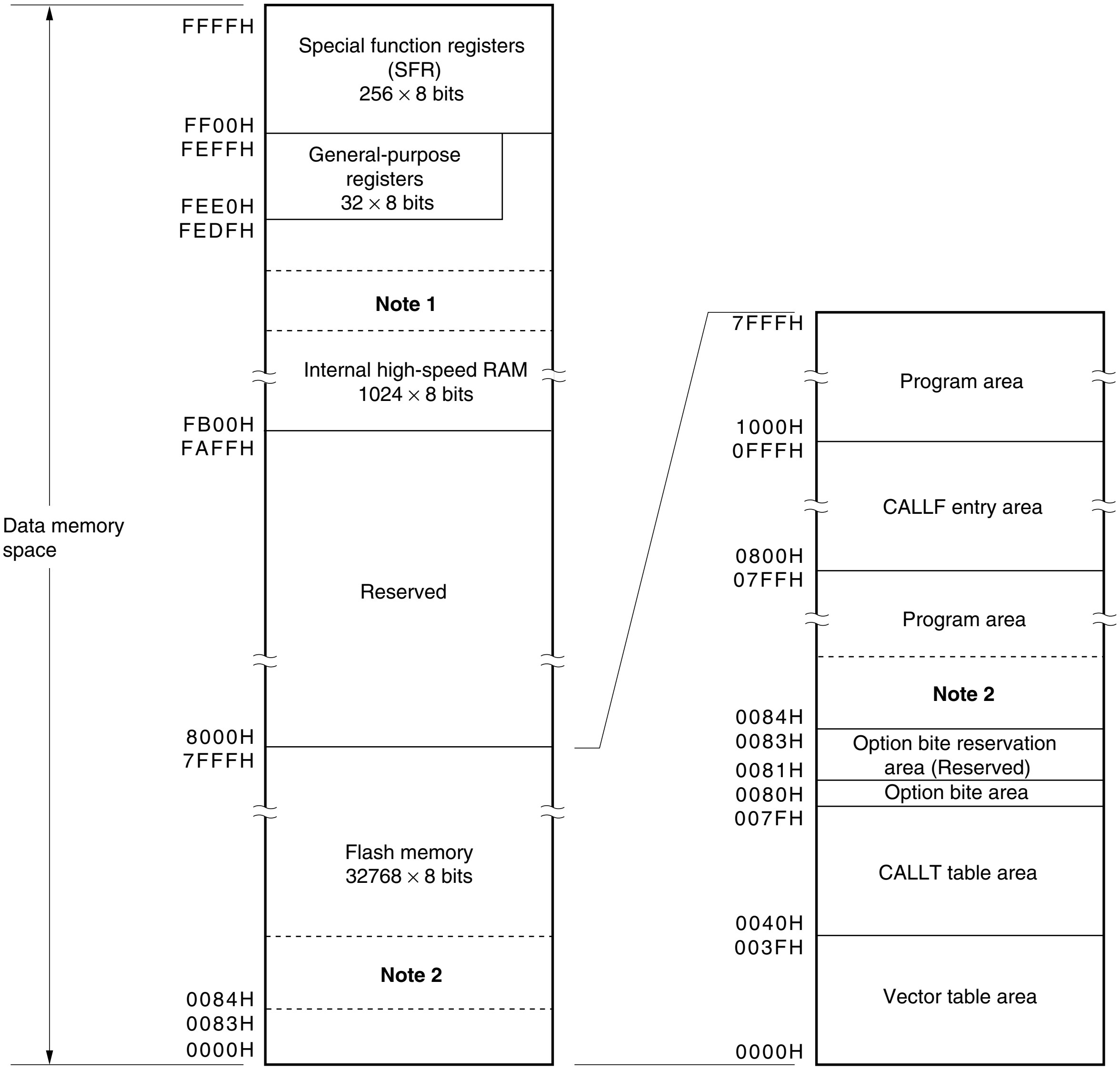
CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

μPD78F0714 products can access a 64 KB memory space. Figures 3-1 shows the memory map.

Caution Because the initial value of the internal memory size switching register (IMS) is CFH, set to C8H by initialization.

Figure 3-1. Memory Map (μPD78F0714)



Notes 1. This area occupies 9 bytes (planned) during on-chip debugging because it is used as a backup area for user data during communication.

2. This area cannot be used during on-chip debugging because it is used as a communication command area (256 bytes to 1 KB).

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

μPD78F0714 products incorporate internal ROM (flash memory), as shown below.

Table 3-1. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F0714	Flash memory	32768 × 8 bits (0000H to 7FFFH)

The internal program memory space is divided into the following areas.

(1) Vector table area

The 64-byte area 0000H to 003FH is reserved as a vector table area. The program start addresses for branch upon reset signal input or generation of each interrupt request are stored in the vector table area.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

Table 3-2. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
0000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT	0020H	INTCM11
		0022H	INTCC10
0004H	INTLVI	0024H	INTCC11
0006H	INTP0	0026H	–Note
0008H	INTP1	0028H	INTTM00
000AH	INTP2	002AH	INTTM01
000CH	INTP3	002CH	INTSRE00
000EH	INTP4	002EH	INTSR00
0010H	INTP5	0030H	INTST00
0012H	INTP6	0032H	INTTM50
0014H	INTP7	0034H	INTTM51
0016H	INTTW0UD	0036H	INTTMH0
0018H	INTTW0CM3	0038H	INTCSI10
001AH	INTTW0CM4	003AH	INTDMU
001CH	INTTW0CM5	003CH	INTAD
001EH	INTCM10		

Note There is no interrupt request corresponding to vector table address 0026H.

(2) CALLT instruction table area

The 64-byte area 0040H to 007FH can store the subroutine entry address of a 1-byte call instruction (CALLT).

(3) Option byte area

The 1-byte area 0080H is reserved as a option byte area. For details, see **CHAPTER 24 OPTION BYTE**.

(4) CALLF instruction entry area

The area 0800H to 0FFFH can perform a direct subroutine call with a 2-byte call instruction (CALLF).

3.1.2 Internal data memory space

μPD78F0714 products incorporate the following RAMs.

(1) Internal high-speed RAM

The internal high-speed RAM is allocated to the area FB00H to FEFFH in a 1024 × 8 bits configuration.

The 32-byte area FEE0H to FEFFH is assigned to four general-purpose register banks consisting of eight 8-bit registers per one bank.

This area cannot be used as a program area in which instructions are written and executed.

The internal high-speed RAM can also be used as a stack memory.

3.1.3 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FF00H to FFFFH (see **Table 3-3 Special Function Register List** in **3.2.3 Special function registers (SFRs)**).

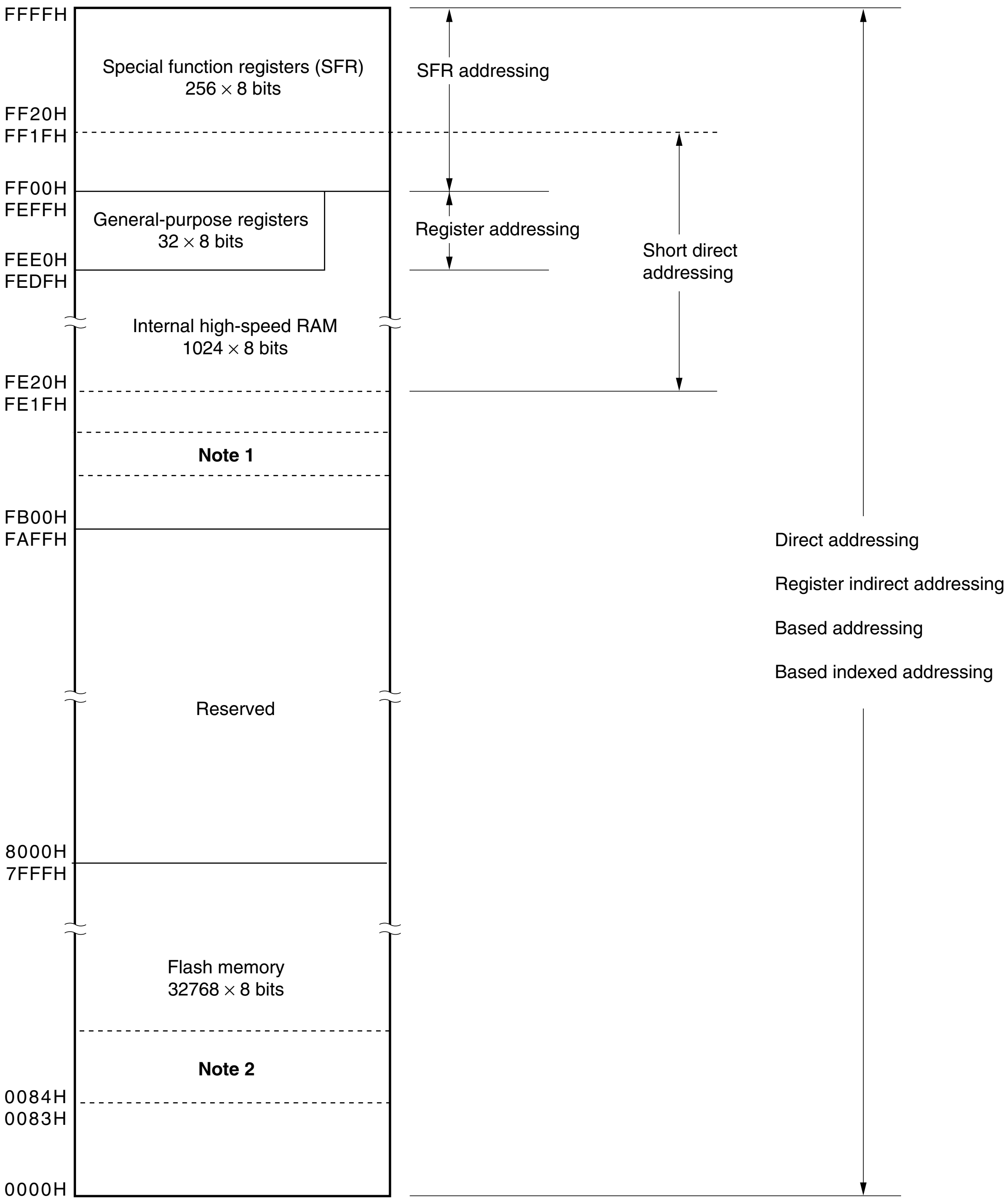
Caution Do not access addresses to which SFRs are not assigned.

3.1.4 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the μPD78F0714, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figure 3-2 shows correspondence between data memory and addressing. For details of each addressing mode, see **3.4 Operand Address Addressing**.

Figure 3-2. Correspondence Between Data Memory and Addressing (μPD78F0714)



Notes 1. This area occupies 9 bytes (planned) during on-chip debugging because it is used as a backup area for user data during communication.

2. This area cannot be used during on-chip debugging because it is used as a communication command area (256 bytes to 1 KB).

3.2 Processor Registers

The μPD78F0714 products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

(1) Program counter (PC)

The program counter is a 16-bit register that holds the address information of the next program to be executed.

In normal operation, the PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

$\overline{\text{RESET}}$ input sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3-3. Format of Program Counter

	15															0
PC	PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are automatically stacked upon interrupt request generation or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

$\overline{\text{RESET}}$ input sets the PSW to 02H.

Figure 3-4. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupts are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

(b) Zero flag (Z)

When the operation result is zero, this flag is set (1). It is reset (0) in all other cases.

(c) Register bank select flags (RBS0 and RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(e) In-service priority flag (ISP)

This flag manages the priority of acknowledgeable maskable vectored interrupts. When this flag is 0, low-level vectored interrupt requests specified by a priority specification flag register (PR0L, PR0H, PR1L, PR1H) (see **19.3 (3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)**) cannot be acknowledged. Actual interrupt request acknowledgment is controlled by the interrupt enable flag (IE).

(f) Carry flag (CY)

This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)

This is a 16-bit register to hold the start address of the memory stack area. Only the internal high-speed RAM area can be set as the stack area.

Figure 3-5. Format of Stack Pointer

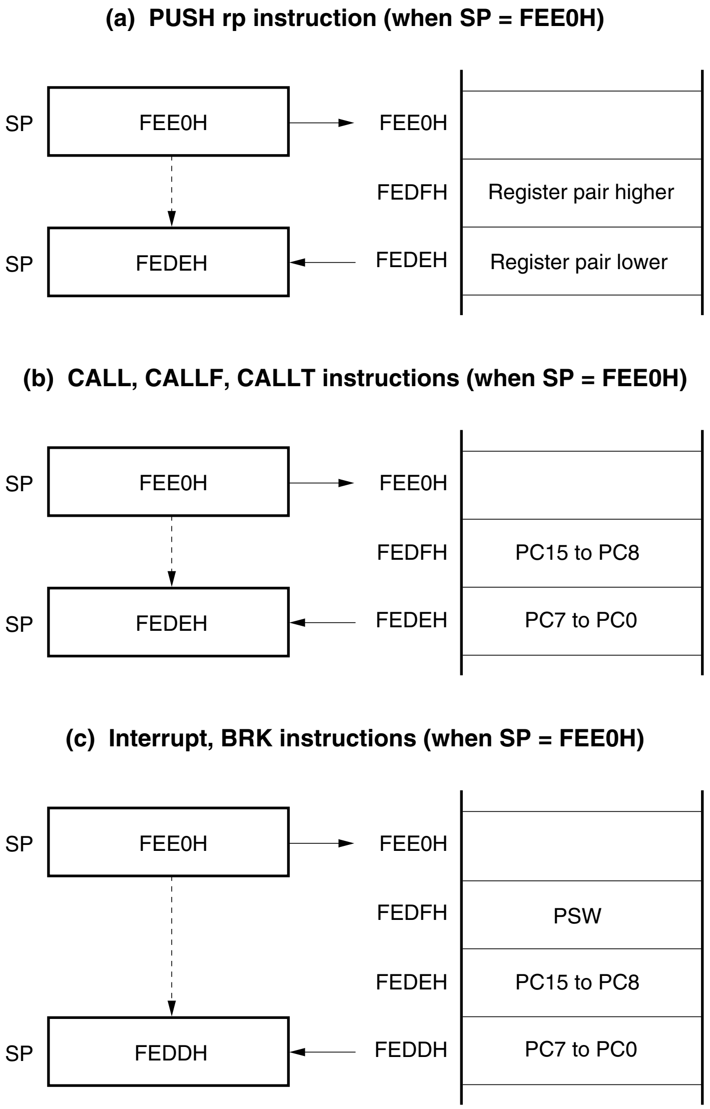
	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

Each stack operation saves/restores data as shown in Figures 3-6 and 3-7.

Caution Since $\overline{\text{RESET}}$ input makes the SP contents undefined, be sure to initialize the SP before using the stack.

Figure 3-6. Data to Be Saved to Stack Memory

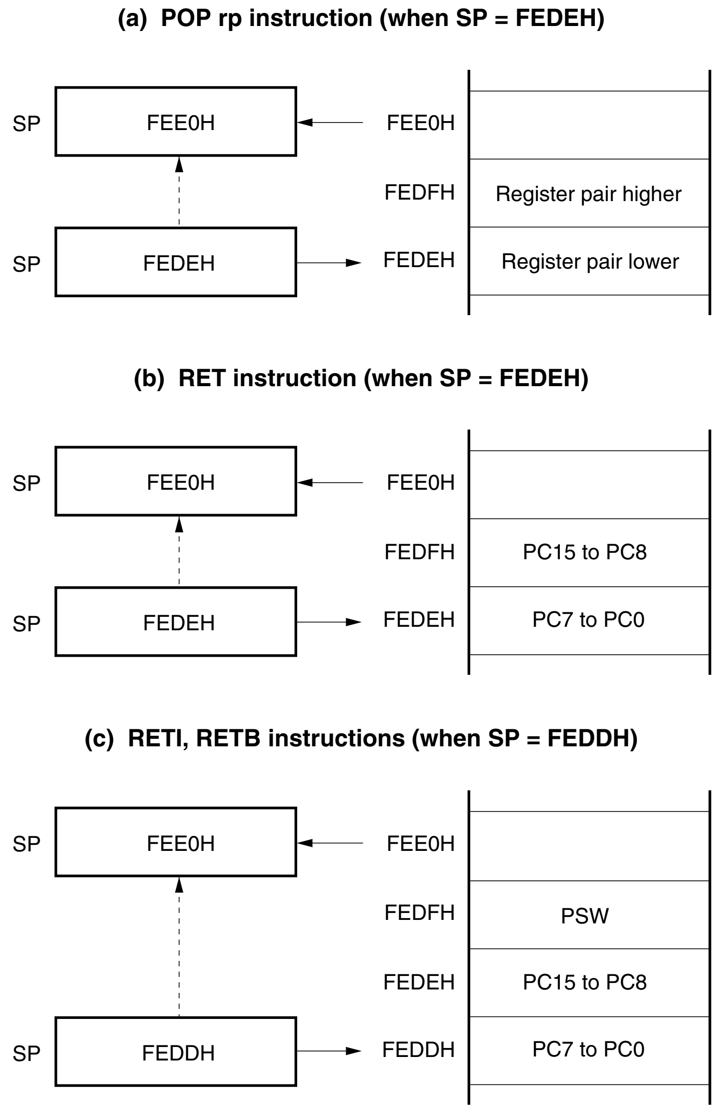


(a) PUSH rp instruction (when SP = FEE0H)

(b) CALL, CALLF, CALLT instructions (when SP = FEE0H)

(c) Interrupt, BRK instructions (when SP = FEE0H)

Figure 3-7. Data to Be Restored from Stack Memory



(a) POP rp instruction (when SP = FEDEH)

(b) RET instruction (when SP = FEDEH)

(c) RETI, RETB instructions (when SP = FEDDH)

3.2.2 General-purpose registers

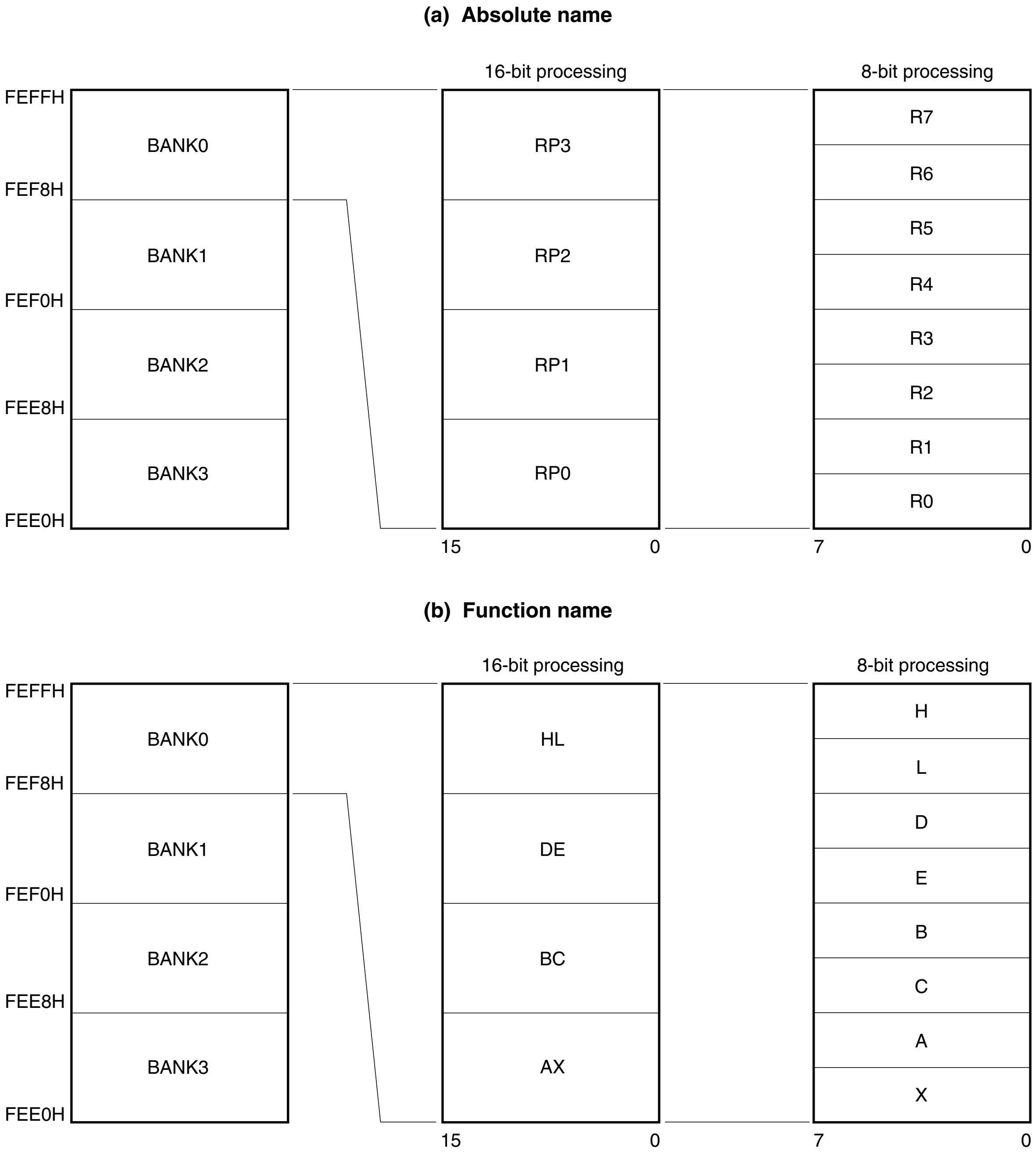
General-purpose registers are mapped at particular addresses (FEE0H to FEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupts for each bank.

Figure 3-8. Configuration of General-Purpose Registers



(a) Absolute name

(b) Function name

3.2.3 Special function registers (SFRs)

Unlike a general-purpose register, each special function register has a special function.

SFRs are allocated to the FF00H to FFFFH area.

Special function registers can be manipulated like general-purpose registers, using operation, transfer and bit manipulation instructions. The manipulatable bit units, 1, 8, and 16, depend on the special function register type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation

  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit).

  This manipulation can also be specified with an address.
- 8-bit manipulation

  Describe the symbol reserved by the assembler for the 8-bit manipulation instruction operand (sfr).

  This manipulation can also be specified with an address.
- 16-bit manipulation

  Describe the symbol reserved by the assembler for the 16-bit manipulation instruction operand (sfrp).

  When specifying an address, describe an even address.

Table 3-3 gives a list of the special function registers. The meanings of items in the table are as follows.

- Symbol

  Symbol indicating the address of a special function register. It is a reserved word in the RA78K0, and is defined as an sfr variable by the #pragma sfr directive for the CC78K0. When using the RA78K0 or ID78K0-QB, symbols can be written as an instruction operand.
- R/W

  Indicates whether the corresponding special function register can be read or written.

  R/W: Read/write enable

  R: Read only

  W: Write only
- Manipulatable bit units

  Indicates the manipulatable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset

  Indicates each register status upon $\overline{\text{RESET}}$ input.

Table 3-3. Special Function Register List (1/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF00H	Port register 0	P0		R/W	√	√	–	00H
FF01H	Port register 1	P1		R/W	√	√	–	00H
FF02H	Port register 2	P2		R	√	√	–	Undefined
FF03H	Port register 3	P3		R/W	√	√	–	00H
FF04H	Port register 4	P4		R/W	√	√	–	00H
FF05H	Port register 5	P5		R/W	√	√	–	00H
FF06H	Port register 6	P6		R/W	√	√	–	00H
FF07H	Port register 7	P7		R/W	√	√	–	00H
FF08H	10-bit buffer register 0	TW0BFCM0	TW0BFCM0L	R/W	–	√	√	0000H
FF09H			–			–		
FF0AH	10-bit buffer register 1	TW0BFCM1	TW0BFCM1L	R/W	–	√	√	0000H
FF0BH			–			–		
FF0CH	10-bit buffer register 2	TW0BFCM2	TW0BFCM2L	R/W	–	√	√	0000H
FF0DH			–			–		
FF0EH	10-bit buffer register 3	TW0BFCM3	TW0BFCM3L	R/W	–	√	√	00FFH
FF0FH			–			–		
FF10H	16-bit up/down counter	IT20UDC	IT20UDCL	R/W	–	√	√	0000H
FF11H			–			–		
FF12H	16-bit compare register 0	IT20CM0	IT20CM0L	R/W	–	√	√	0000H
FF13H			–			–		
FF14H	16-bit compare register 1	IT20CM1	IT20CM1L	R/W	–	√	√	0000H
FF15H			–			–		
FF16H	16-bit timer counter 00	TM00		R	–	–	√	0000H
FF17H								
FF18H	Receive buffer register 0	RXB00		R	–	√	–	FFH
FF19H	Transmit shift register 0	TXS00		W	–	√	–	FFH
FF1AH	A/D conversion result register	ADCR		R	–	–	√	Undefined
FF1BH								
FF1FH	Serial I/O shift register 10	SIO10		R	–	√	–	00H
FF20H	Port mode register 0	PM0		R/W	√	√	–	FFH
FF21H	Port mode register 1	PM1		R/W	√	√	–	FFH
FF23H	Port mode register 3	PM3		R/W	√	√	–	FFH
FF24H	Port mode register 4	PM4		R/W	√	√	–	FFH
FF25H	Port mode register 5	PM5		R/W	√	√	–	FFH
FF26H	Port mode register 6	PM6		R/W	√	√	–	FFH
FF27H	Port mode register 7	PM7		R/W	√	√	–	FFH

Table 3-3. Special Function Register List (2/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				1 Bit	8 Bits	16 Bits	
FF28H	DC control register 00	DCCTL00	R/W	√	√	–	00H
FF2AH	8-bit timer H mode register 0	TMHMD0	R/W	√	√	–	00H
FF2CH	8-bit timer counter 50	TM50	R	–	√	–	00H
FF2DH	8-bit timer compare register 50	CR50	R/W	–	√	–	00H
FF2EH	Timer clock selection register 50	TCL50	R/W	–	√	–	00H
FF2FH	8-bit timer mode control register 50	TMC50	R/W	√	√	–	00H
FF30H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
FF31H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
FF33H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
FF34H	Pull-up resistor option register 4	PU4	R/W	√	√	–	00H
FF35H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
FF36H	Pull-up resistor option register 6	PU6	R/W	√	√	–	00H
FF37H	Pull-up resistor option register 7	PU7	R/W	√	√	–	00H
FF38H	DC control register 01	DCCTL01	R/W	√	√	–	00H
FF3AH	Prescaler mode register	IT20PRM	R/W	√	√	–	07H
FF3BH	Status register	IT20STS	R	√	√	–	00H
FF3CH	8-bit timer counter 51	TM51	R	–	√	–	00H
FF3DH	8-bit timer compare register 51	CR51	R/W	–	√	–	00H
FF3EH	Timer clock selection register 51	TCL51	R/W	–	√	–	00H
FF3FH	8-bit timer mode control register 51	TMC51	R/W	√	√	–	00H
FF40H	Clock output selection register	CKS	R/W	√	√	–	00H
FF48H	External interrupt rising edge enable register	EGP	R/W	√	√	–	00H
FF49H	External interrupt falling edge enable register	EGN	R/W	√	√	–	00H
FF50H	10-bit buffer register 4	TW0BFCM4 TW0BFCM4L	R/W	–	√	√	0000H
FF51H		–			–		
FF52H	10-bit buffer register 5	TW0BFCM5 TW0BFCM5L	R/W	–	√	√	0000H
FF53H		–			–		
FF54H	10-bit compare register 0	TW0CM0	R/W	–	–	√	0000H
FF55H							
FF56H	10-bit compare register 1	TW0CM1	R/W	–	–	√	0000H
FF57H							
FF58H	10-bit compare register 2	TW0CM2	R/W	–	–	√	0000H
FF59H							
FF5AH	10-bit compare register 3	TW0CM3	R/W	–	–	√	00FFH
FF5BH							
FF5CH	10-bit compare register 4	TW0CM4	R/W	–	–	√	0000H
FF5DH							
FF5EH	10-bit compare register 5	TW0CM5	R/W	–	–	√	0000H
FF5FH							

Table 3-3. Special Function Register List (3/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After Reset
					1 Bit	8 Bits	16 Bits	
FF60H	Remainder data register 0	SDR0	SDR0L	R	–	√	√	00H
FF61H			SDR0H		–	√		00H
FF62H	Multiplication/division data register A0	MDA0L	MDA0LL	R/W	–	√	√	00H
FF63H			MDA0LH		–	√		00H
FF64H		MDA0H	MDA0HL	R/W	–	√	√	00H
FF65H			MDA0HH		–	√		00H
FF66H	Multiplication/division data register B0	MDB0	MDB0L	R/W	–	√	√	00H
FF67H			MDB0H		–	√		00H
FF68H	Multiplier/divider control register 0	DMUC0		R/W	√	√	–	00H
FF6AH	Capture/compare control register 00	CRC00		R/W	√	√	–	00H
FF6BH	16-bit timer output control register 00	TOC00		R/W	√	√	–	00H
FF6CH	A/D converter mode register	ADM		R/W	√	√	–	00H
FF6DH	Analog input channel specification register	ADS		R/W	√	√	–	00H
FF6EH	Power-fail comparison mode register	PFM		R/W	√	√	–	00H
FF6FH	Power-fail comparison threshold register	PFT		R/W	–	√	–	00H
FF70H	Asynchronous serial interface operation mode register 00	ASIM00		R/W	√	√	–	01H
FF71H	Baud rate generator control register 00	BRGC00		R/W	–	√	–	1FH
FF73H	Asynchronous serial interface reception error status register 00	ASIS00		R	–	√	–	00H
FF78H	Low-voltage detection register	LVIM		R/W	√	√	–	00H$^{\text{Note}}$
FF7AH	16-bit timer capture/compare register 00	CR00		R/W	–	–	√	0000H
FF7BH								
FF7CH	16-bit timer capture/compare register 01	CR01		R/W	–	–	√	0000H
FF7DH								
FF7EH	16-bit timer mode control register 00	TMC00		R/W	√	√	–	00H
FF7FH	Prescaler mode register 00	PRM00		R/W	√	√	–	00H
FF80H	Serial operation mode register 10	CSIM10		R/W	√	√	–	00H
FF81H	Serial clock selection register 10	CSIC10		R/W	√	√	–	00H
FF84H	Transmit buffer register 10	SOTB10		R/W	–	√	–	Undefined
FF88H	Inverter timer control register	TW0C		R/W	√	√	–	00H
FF89H	Inverter timer mode register	TW0M		R/W	√	√	–	00H
FF8AH	Dead time reload register	TW0DTIME		R/W	–	√	–	FFH
FF8BH	A/D trigger select register	TW0TRGS		R/W	√	√	–	00H
FF8CH	Inverter timer output control register	TW0OC		R/W	√	√	–	00H

Note This value is 83H only after a LVI reset.

Table 3-3. Special Function Register List (4/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After
					1 Bit	8 Bits	16 Bits	Reset
FF90H	16-bit timer capture/compare register 0	IT20C C0	IT20C C0L	R/W	–	√	√	0000H
FF91H			–			–		
FF92H	16-bit timer capture/compare register 1	IT20C C1	IT20C C1L	R/W	–	√	√	0000H
FF93H			–			–		
FF94H	Capture/compare control register	IT20CCR		R/W	√	√	–	00H
FF95H	Timer unit mode register	IT20TUM		R/W	√	√	–	00H
FF96H	Timer control register	IT20TMC		R/W	√	√	–	00H
FF97H	Effective edge select register	IT20SESA		R/W	√	√	–	00H
FF98H	Watchdog timer mode register	WDTM		R/W	–	√	–	67H
FF99H	Watchdog timer enable register	WDTE		R/W	–	√	–	9AH
FF9EH	8-bit timer H compare register 00	CMP00		R/W	–	√	–	00H
FF9FH	8-bit timer H compare register 01	CMP01		R/W	–	√	–	00H
FFA0H	Internal oscillation mode register	RCM		R/W	√	√	–	00H
FFA1H	Main clock mode register	MCM		R/W	√	√	–	00H
FFA2H	Main OSC control register	MOC		R/W	√	√	–	00H
FFA3H	Oscillation stabilization time counter status register	OSTC		R	√	√	–	00H
FFA4H	Oscillation stabilization time select register	OSTS		R/W	–	√	–	05H
FFAAH	Noise eliminate time select register	NRC1		R/W	√	√	–	00H
FFACH	Reset control flag register	RESF		R	–	√	–	00H[Note 1]
FFB0H	Real-time output buffer register 0L	RTBL00		R/W	√	√	–	00H
FFB2H	Real-time output buffer register 0H	RTBH00		R/W	√	√	–	00H
FFB4H	Real-time output port mode register 0	RTPM00		R/W	√	√	–	00H
FFB5H	Real-time output port control register 0	RTPC00		R/W	√	√	–	00H
FFB8H	Real-time output buffer register 1L	RTBL01		R/W	√	√	–	00H
FFBAH	Real-time output buffer register 1H	RTBH01		R/W	√	√	–	00H
FFBCH	Real-time output port mode register 1	RTPM01		R/W	√	√	–	00H
FFBDH	Real-time output port control register 1	RTPC01		R/W	√	√	–	00H
FFC0H	Flash protect command register	PFCMD		W	–	√	–	Undefined
FFC2H	Flash status register	PFS		R/W	√	√	–	00H
FFC4H	Flash programming mode control register	FLPMC		R/W	√	√	–	0XH[Note 2]

Notes 1. This value varies depending on the reset source.

2. Differs depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

Table 3-3. Special Function Register List (5/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Unit			After
					1 Bit	8 Bits	16 Bits	Reset
FFE0H	Interrupt request flag register 0L	IF0	IF0L	R/W	√	√	√	00H
FFE1H	Interrupt request flag register 0H		IF0H	R/W	√	√		00H
FFE2H	Interrupt request flag register 1L	IF1	IF1L	R/W	√	√	√	00H
FFE3H	Interrupt request flag register 1H		IF1H	R/W	√	√		00H
FFE4H	Interrupt mask flag register 0L	MK0	MK0L	R/W	√	√	√	FFH
FFE5H	Interrupt mask flag register 0H		MK0H	R/W	√	√		FFH
FFE6H	Interrupt mask flag register 1L	MK1	MK1L	R/W	√	√	√	FFH
FFE7H	Interrupt mask flag register 1H		MK1H	R/W	√	√		DFH
FFE8H	Priority specification flag register 0L	PR0	PR0L	R/W	√	√	√	FFH
FFE9H	Priority specification flag register 0H		PR0H	R/W	√	√		FFH
FFEAH	Priority specification flag register 1L	PR1	PR1L	R/W	√	√	√	FFH
FFEBH	Priority specification flag register 1H		PR1H	R/W	√	√		FFH
FFF0H	Internal memory size switching register [Note]	IMS		R/W	–	√	–	CFH
FFFBH	Processor clock control register	PCC		R/W	√	√	–	00H
FFFDH	System wait control register	VSWC		R/W	√	√	–	00H

Note Because the initial value of the internal memory size switching register (IMS) is CFH, set to C8H by initialization.

3.3 Instruction Address Addressing

An instruction address is determined by program counter (PC) contents and is normally incremented (+1 for each byte) automatically according to the number of bytes of an instruction to be fetched each time another instruction is executed. When a branch instruction is executed, the branch destination information is set to the PC and branched by the following addressing (for details of instructions, refer to **78K/0 Series Instructions User's Manual (U12326E)**).

3.3.1 Relative addressing

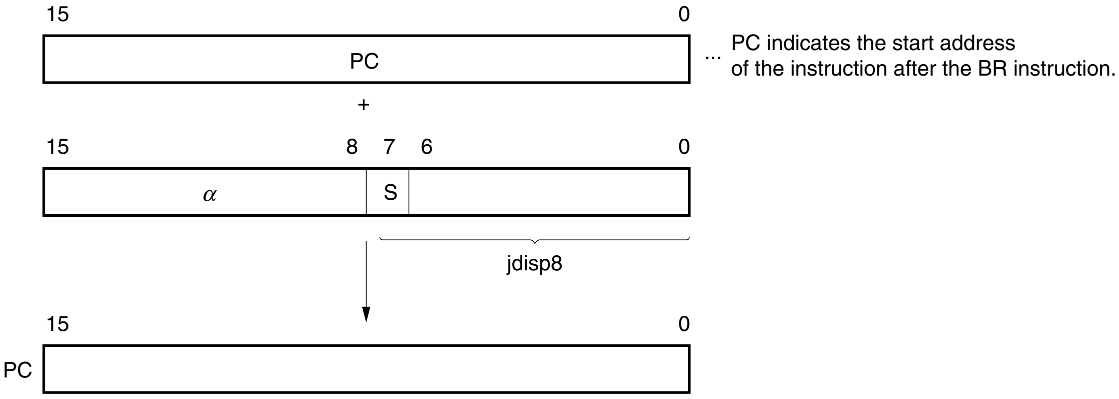
[Function]

The value obtained by adding 8-bit immediate data (displacement value: jdisp8) of an instruction code to the start address of the following instruction is transferred to the program counter (PC) and branched. The displacement value is treated as signed two's complement data (–128 to +127) and bit 7 becomes a sign bit.

In other words, relative addressing consists of relative branching from the start address of the following instruction to the –128 to +127 range.

This function is carried out when the BR $addr16 instruction or a conditional branch instruction is executed.

[Illustration]


15
0
PC
... PC indicates the start address of the instruction after the BR instruction.
+
15
8
7
6
0
α
S
jdisp8
15
0
PC

When S = 0, all bits of α are 0.
When S = 1, all bits of α are 1.

3.3.2 Immediate addressing

[Function]

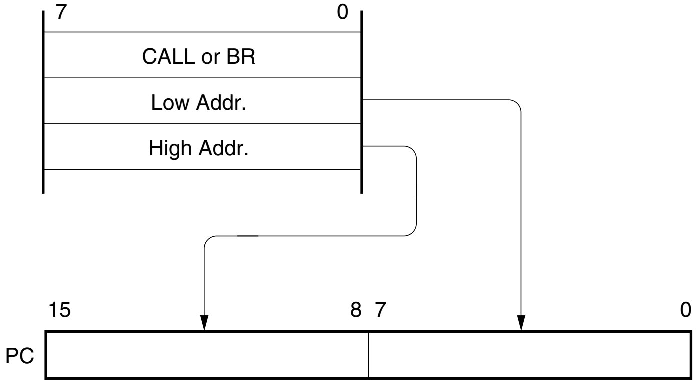
Immediate data in the instruction word is transferred to the program counter (PC) and branched.

This function is carried out when the CALL !addr16 or BR !addr16 or CALLF !addr11 instruction is executed.

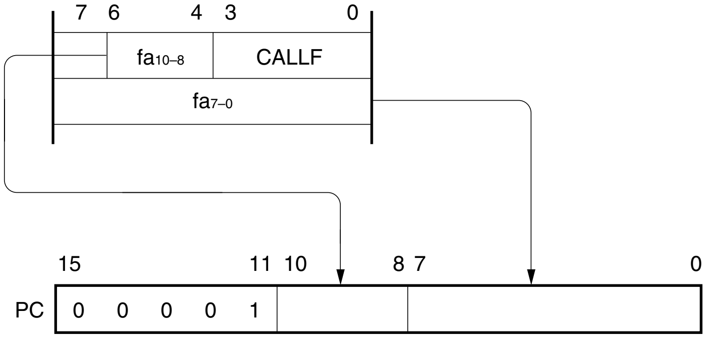
CALL !addr16 and BR !addr16 instructions can be branched to the entire memory space. The CALLF !addr11 instruction is branched to the 0800H to 0FFFH area.

[Illustration]

In the case of CALL !addr16 and BR !addr16 instructions



In the case of CALLF !addr11 instruction



<R>

3.3.3 Table indirect addressing

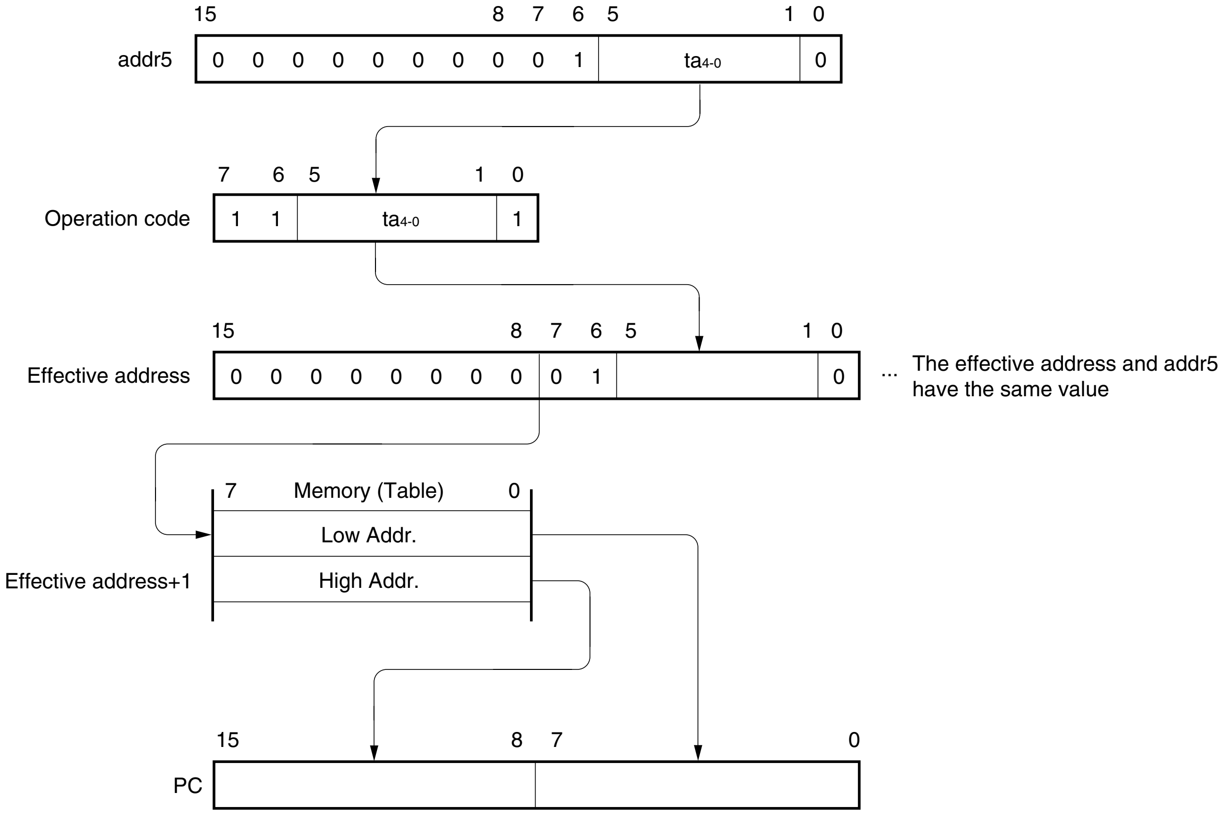
[Function]

Table contents (branch destination address) of the particular location to be addressed by bits 1 to 5 of the immediate data of an operation code are transferred to the program counter (PC) and branched.

This function is carried out when the CALLT [addr5] instruction is executed.

This instruction references the address stored in the memory table from 0040H to 007FH, which is indicated by addr5, and allows branching to the entire memory space.

[Illustration]



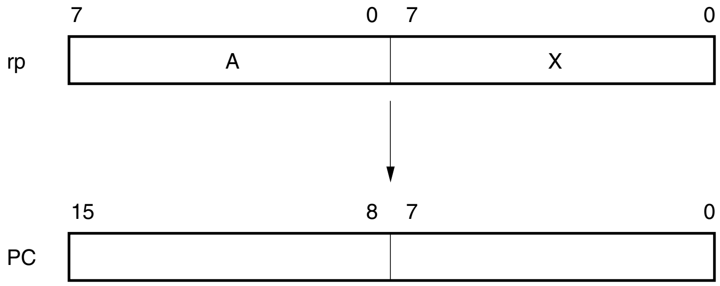
3.3.4 Register addressing

[Function]

Register pair (AX) contents to be specified with an instruction word are transferred to the program counter (PC) and branched.

This function is carried out when the BR AX instruction is executed.

[Illustration]



3.4 Operand Address Addressing

The following methods are available to specify the register and memory (addressing) to undergo manipulation during instruction execution.

3.4.1 Implied addressing

[Function]

The register that functions as an accumulator (A and AX) among the general-purpose registers is automatically (implicitly) addressed.

Of the μPD78F0714 instruction words, the following instructions employ implied addressing.

Instruction	Register to Be Specified by Implied Addressing
MULU	A register for multiplicand and AX register for product storage
DIVUW	AX register for dividend and quotient storage
ADJBA/ADJBS	A register for storage of numeric values that become decimal correction targets
ROR4/ROL4	A register for storage of digit data that undergoes digit rotation

[Operand format]

Because implied addressing can be automatically employed with an instruction, no particular operand format is necessary.

[Description example]

In the case of MULU X

With an 8-bit × 8-bit multiply instruction, the product of A register and X register is stored in AX. In this example, the A and AX registers are specified by implied addressing.

3.4.2 Register addressing

[Function]

The general-purpose register to be specified is accessed as an operand with the register bank select flags (RBS0 to RBS1) and the register specify codes (Rn and RPn) of an operation code.

Register addressing is carried out when an instruction with the following operand format is executed. When an 8-bit register is specified, one of the eight registers is specified with 3 bits in the operation code.

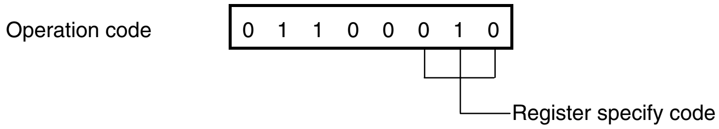
[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

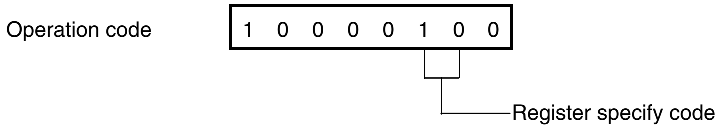
'r' and 'rp' can be described by absolute names (R0 to R7 and RP0 to RP3) as well as function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL).

[Description example]

MOV A, C; when selecting C register as r



INCW DE; when selecting DE register pair as rp



3.4.3 Direct addressing

[Function]

The memory to be manipulated is directly addressed with immediate data in an instruction word becoming an operand address.

[Operand format]

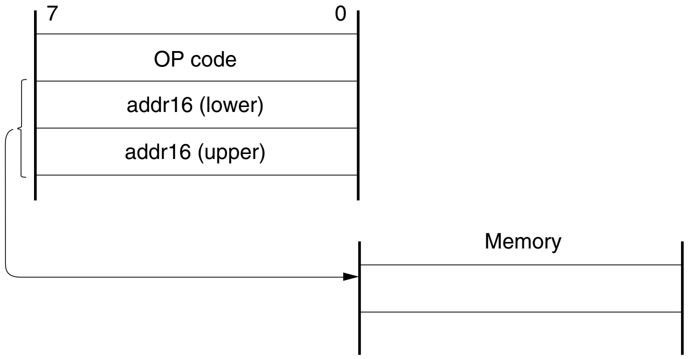
Identifier	Description
addr16	Label or 16-bit immediate data

[Description example]

MOV A, !0FE00H; when setting !addr16 to FE00H

Operation code	1 0 0 0 1 1 1 0	OP code
	0 0 0 0 0 0 0 0	00H
	1 1 1 1 1 1 1 0	FEH

[Illustration]



3.4.4 Short direct addressing

[Function]

The memory to be manipulated in the fixed space is directly addressed with 8-bit data in an instruction word.

This addressing is applied to the 256-byte space FE20H to FF1FH. Internal RAM and special function registers (SFRs) are mapped at FE20H to FEFFH and FF00H to FF1FH, respectively.

The SFR area (FF00H to FF1FH) where short direct addressing is applied is a part of the overall SFR area. Ports that are frequently accessed in a program and compare and capture registers of the timer/event counter are mapped in this area, allowing SFRs to be manipulated with a small number of bytes and clocks.

When 8-bit immediate data is at 20H to FFH, bit 8 of an effective address is cleared to 0. When it is at 00H to 1FH, bit 8 is set to 1. Refer to the **[Illustration]**.

[Operand format]

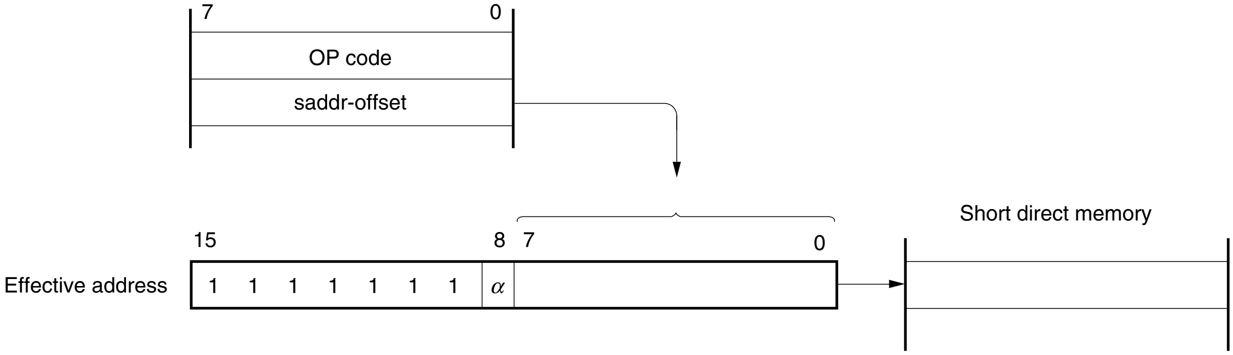
Identifier	Description
saddr	Immediate data that indicate label or FE20H to FF1FH
saddrp	Immediate data that indicate label or FE20H to FF1EH (even address only)

[Description example]

MOV 0FE30H, A; when transferring value of A register to saddr (FE30H)

Operation code	1 1 1 1 0 0 1 0	OP code
	0 0 1 1 0 0 0 0	30H (saddr-offset)

[Illustration]



When 8-bit immediate data is 20H to FFH, $\alpha = 0$

When 8-bit immediate data is 00H to 1FH, $\alpha = 1$

3.4.5 Special function register (SFR) addressing

[Function]

A memory-mapped special function register (SFR) is addressed with 8-bit immediate data in an instruction word. This addressing is applied to the 240-byte spaces FF00H to FFCFH and FFE0H to FFFFH. However, the SFRs mapped at FF00H to FF1FH can be accessed with short direct addressing.

[Operand format]

Identifier	Description
sfr	Special function register name
sfrp	16-bit manipulatable special function register name (even address only)

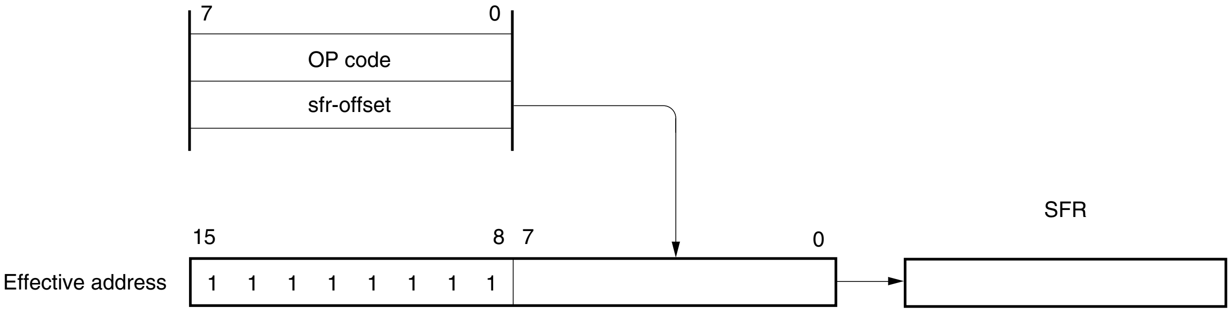
[Description example]

MOV PM0, A; when selecting PM0 (FF20H) as sfr

Operation code 1 1 1 1 0 1 1 0 OP code

0 0 1 0 0 0 0 0 20H (sfr-offset)

[Illustration]



3.4.6 Register indirect addressing

[Function]

Register pair contents specified by a register pair specify code in an instruction word and by a register bank select flag (RBS0 and RBS1) serve as an operand address for addressing the memory. This addressing can be carried out for all the memory spaces.

[Operand format]

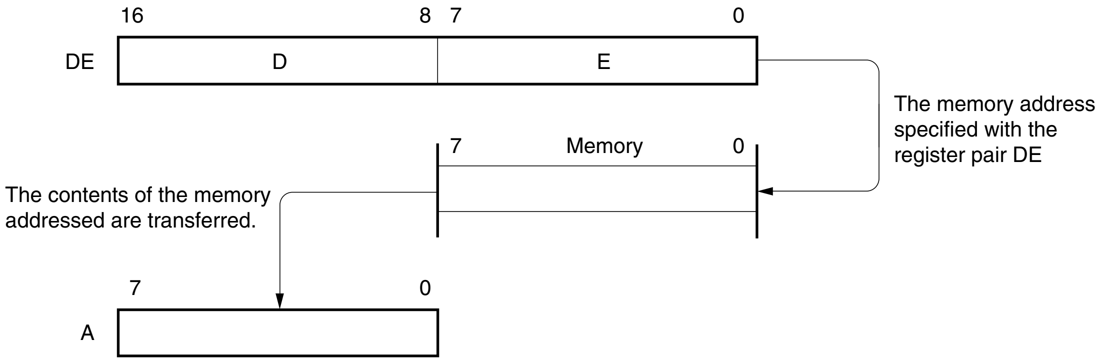
Identifier	Description
–	[DE], [HL]

[Description example]

MOV A, [DE]; when selecting [DE] as register pair

Operation code 1 0 0 0 0 1 0 1

[Illustration]


16
8 7
0
DE
D
E
The memory address specified with the register pair DE
7
Memory
0
The contents of the memory addressed are transferred.
7
0
A

3.4.7 Based addressing

[Function]

8-bit immediate data is added as offset data to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the offset data as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

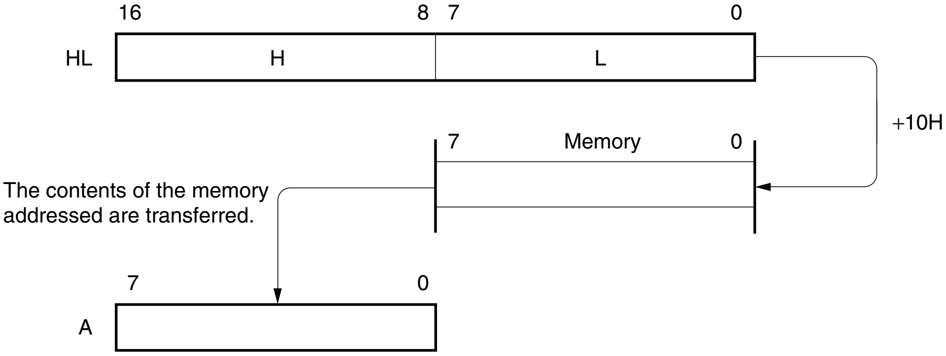
Identifier	Description
–	[HL + byte]

[Description example]

MOV A, [HL + 10H]; when setting byte to 10H

Operation code	1	0	1	0	1	1	1	0
	0	0	0	1	0	0	0	0

[Illustration]



3.4.8 Based indexed addressing

[Function]

The B or C register contents specified in an instruction word are added to the contents of the base register, that is, the HL register pair in the register bank specified by the register bank select flag (RBS0 and RBS1), and the sum is used to address the memory. Addition is performed by expanding the B or C register contents as a positive number to 16 bits. A carry from the 16th bit is ignored. This addressing can be carried out for all the memory spaces.

[Operand format]

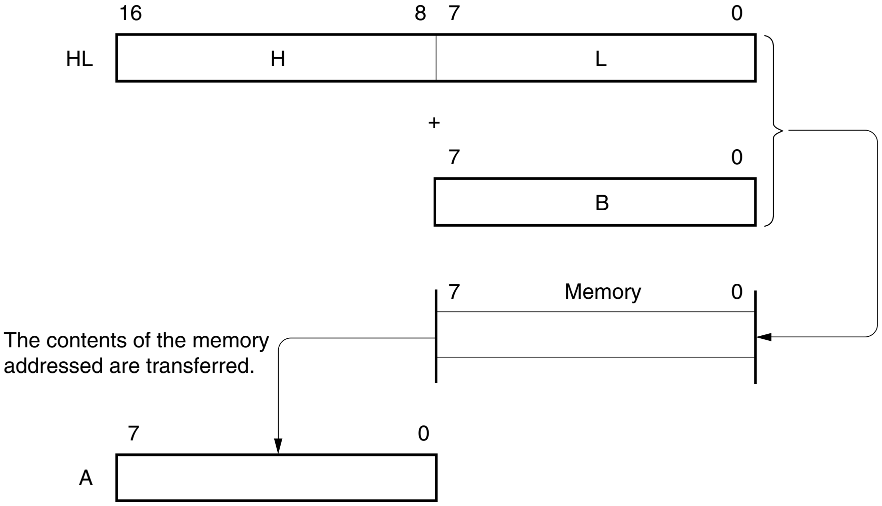
Identifier	Description
–	[HL + B], [HL + C]

[Description example]

In the case of MOV A, [HL + B] (selecting B register)

Operation code	1	0	1	0	1	0	1	1

[Illustration]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents.

This addressing method is automatically employed when the PUSH, POP, subroutine call and return instructions are executed or the register is saved/reset upon generation of an interrupt request.

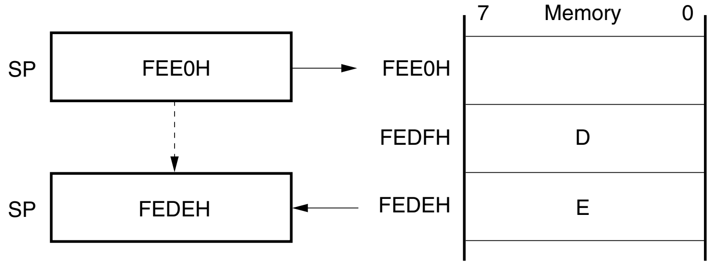
With stack addressing, only the internal high-speed RAM area can be accessed.

[Description example]

In the case of PUSH DE (saving DE register)

Operation code 1 0 1 1 0 1 0 1

[Illustration]



CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are two types of pin I/O buffer power supplies: AV_{REF} and EV_{DD}. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{REF}	P20 to P27[Note]
EV_{DD}	Port pins other than P20 to P27

Note Connect AV_{REF} to EV_{DD} when port 2 is used as a digital port.

μPD78F0714 products are provided with the ports shown in Figure 4-1, which enable variety of control operations. The functions of each port are shown in Table 4-2.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

Figure 4-1. Port Types

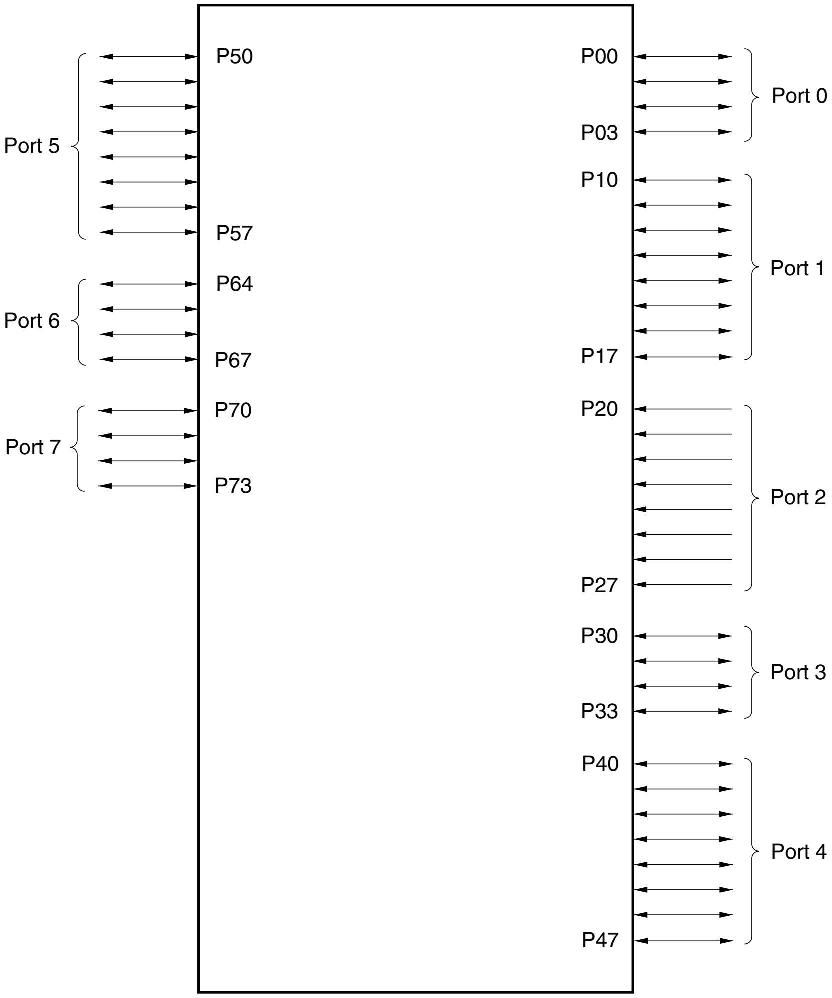


Table 4-2. Port Functions

Pin Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	INTP0/TW0TOFFP
P01				INTP1
P02				INTP2
P03				INTP3/ADTRG
P10	I/O	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–
P11				–
P12				–
P13				RxD00
P14				TxD00
P15				$\overline{\text{SCK10}}$
P16				SI10
P17				SO10/FLMD1
P20 to P27	Input	Port 2. 8-bit input-only port.	Input	ANI0 to ANI7
P30	I/O	Port 3. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	BUZ
P31				PCL
P32				–
P33				–
P40 to P47	I/O	Port 4. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	RTP00 to RTP07
P50	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	TI50/TO50
P51				TI51/TO51
P52				TOH0/INTP4
P53				TI000/INTP5
P54				TI001/TO00
P55				TIT20IUD/INTP6
P56				TIT20CUD /TIT20CC0/INTP7
P57				TIT20CLR /TIT20CC1 /TIT20TO
P64 to P67	I/O	Port 6. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–
P70 to P73	I/O	Port 7. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input	–

4.2 Port Configuration

Ports consist of the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode register (PM0, PM1, PM3 to PM7) Port register (P0 to P7) Pull-up resistor option register (PU0, PU1, PU3 to PU7)
Port	Total: 48 (CMOS I/O: 40, CMOS input: 8)
Pull-up resistor	Total: 40 (software control: 40)

4.2.1 Port 0

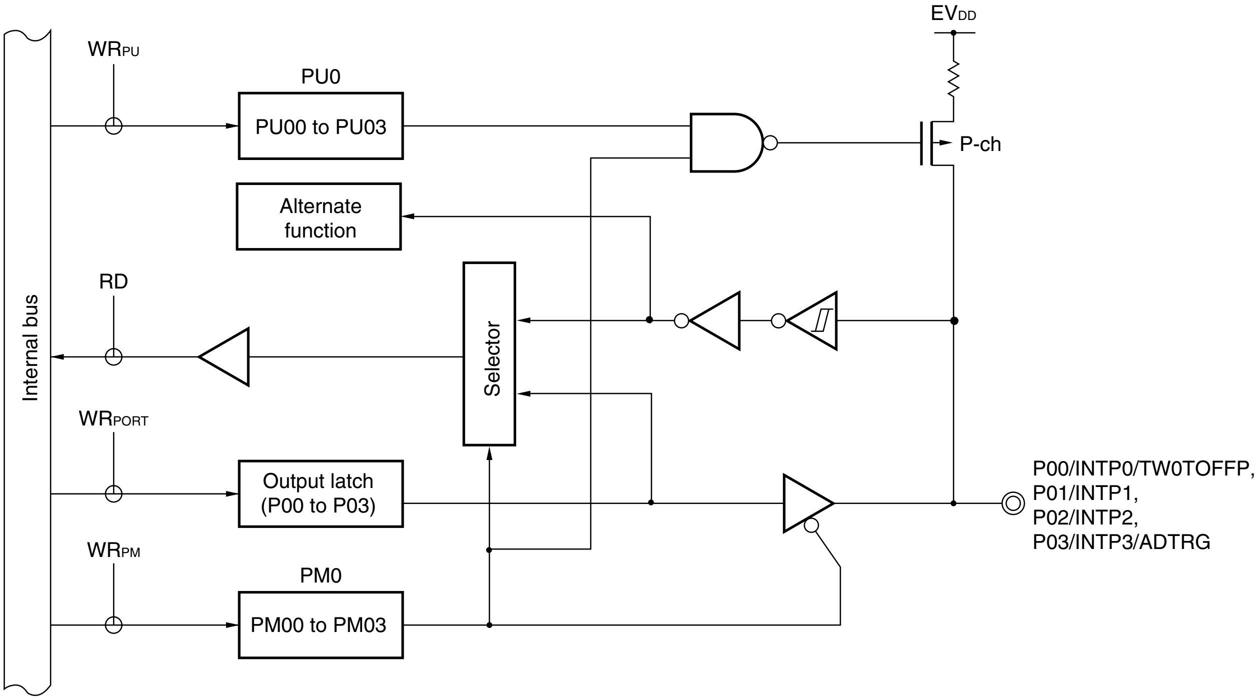
Port 0 is a 4-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P03 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

This port can also be used for external interrupt request input, timer output stop external signal, and A/D converter trigger input.

$\overline{\text{RESET}}$ input sets port 0 to input mode.

Figure 4-2 shows a block diagram of port 0.

Figure 4-2. Block Diagram of P00 to P03



PU0: Pull-up resistor option register 0

PM0: Port mode register 0

RD: Read signal

WR××: Write signal

4.2.2 Port 1

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

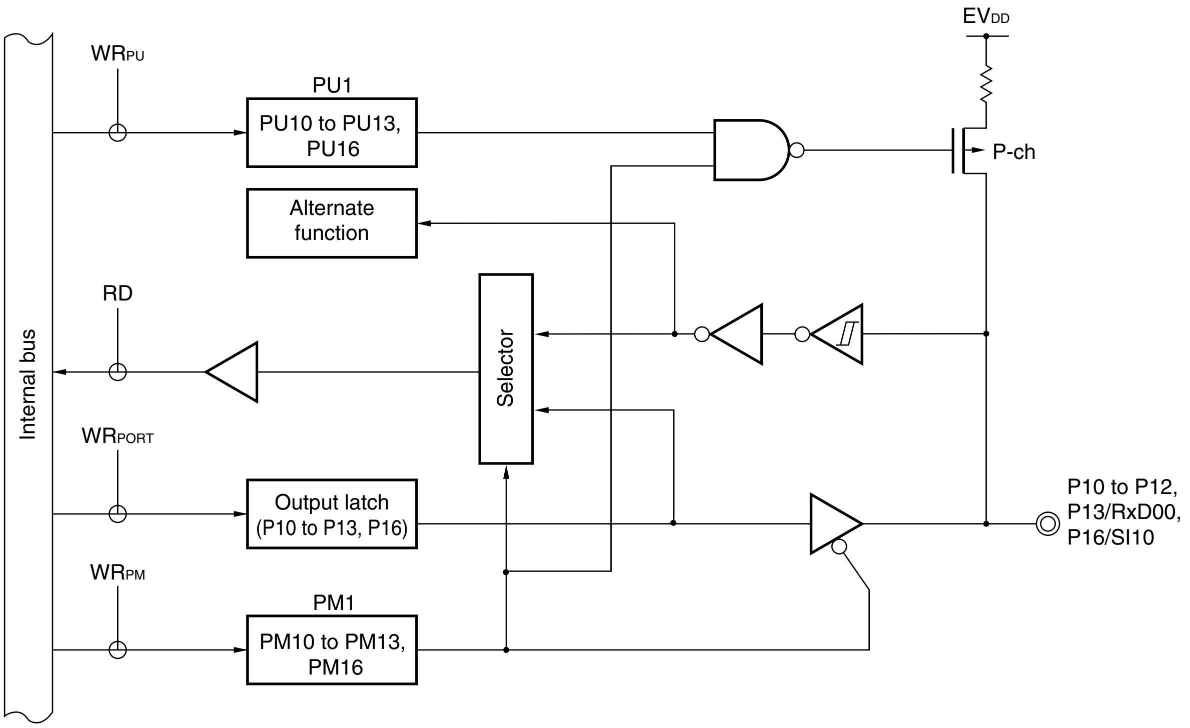
This port can also be used for serial interface data I/O, clock I/O, and flash memory programming mode setting.

$\overline{\text{RESET}}$ input sets port 1 to input mode.

Figures 4-3 to 4-6 show block diagrams of port 1.

Caution When P15/$\overline{\text{SCK10}}$, P16/SI10, and P17/SO10 are used as general-purpose ports, do not write to serial clock selection register 10 (CSIC10).

Figure 4-3. Block Diagram of P10 toP13 and P16



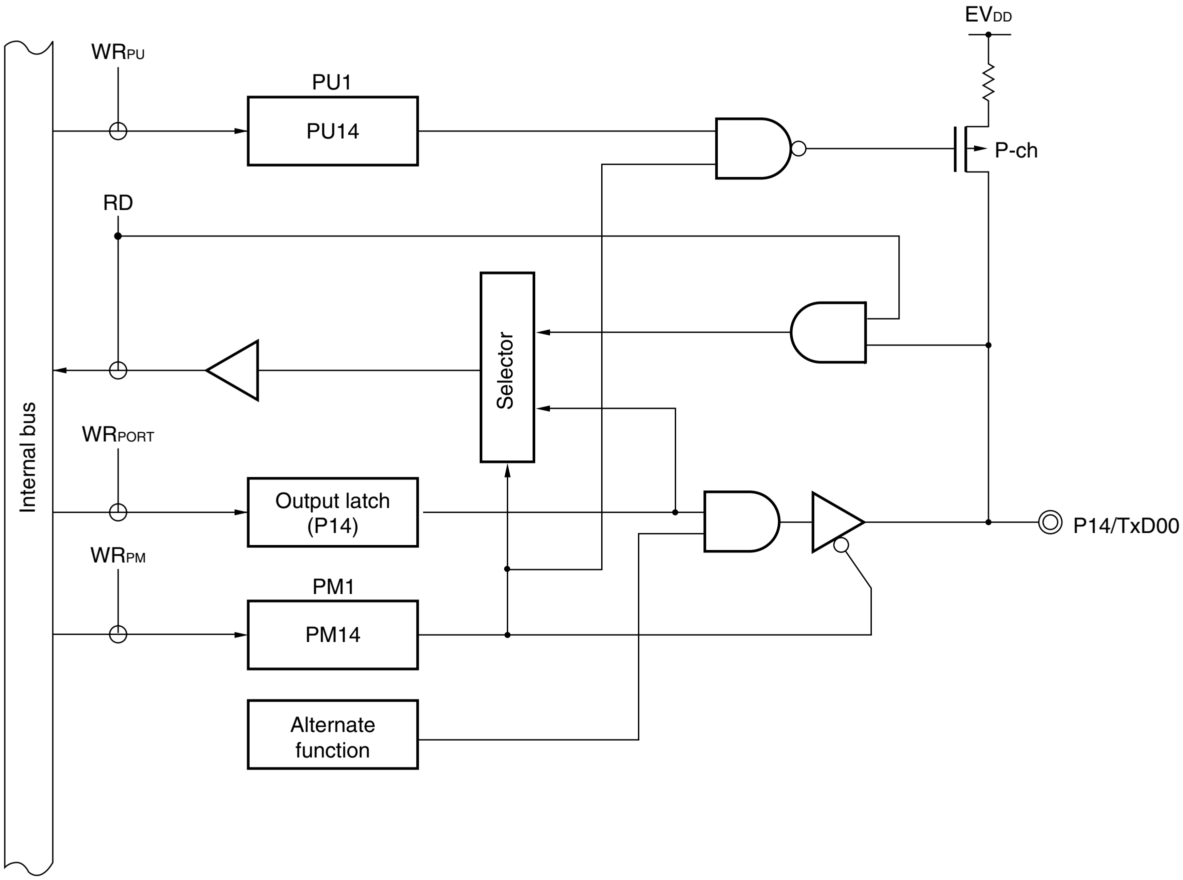
PU1: Pull-up resistor option register 1

PM1: Port mode register 1

RD: Read signal

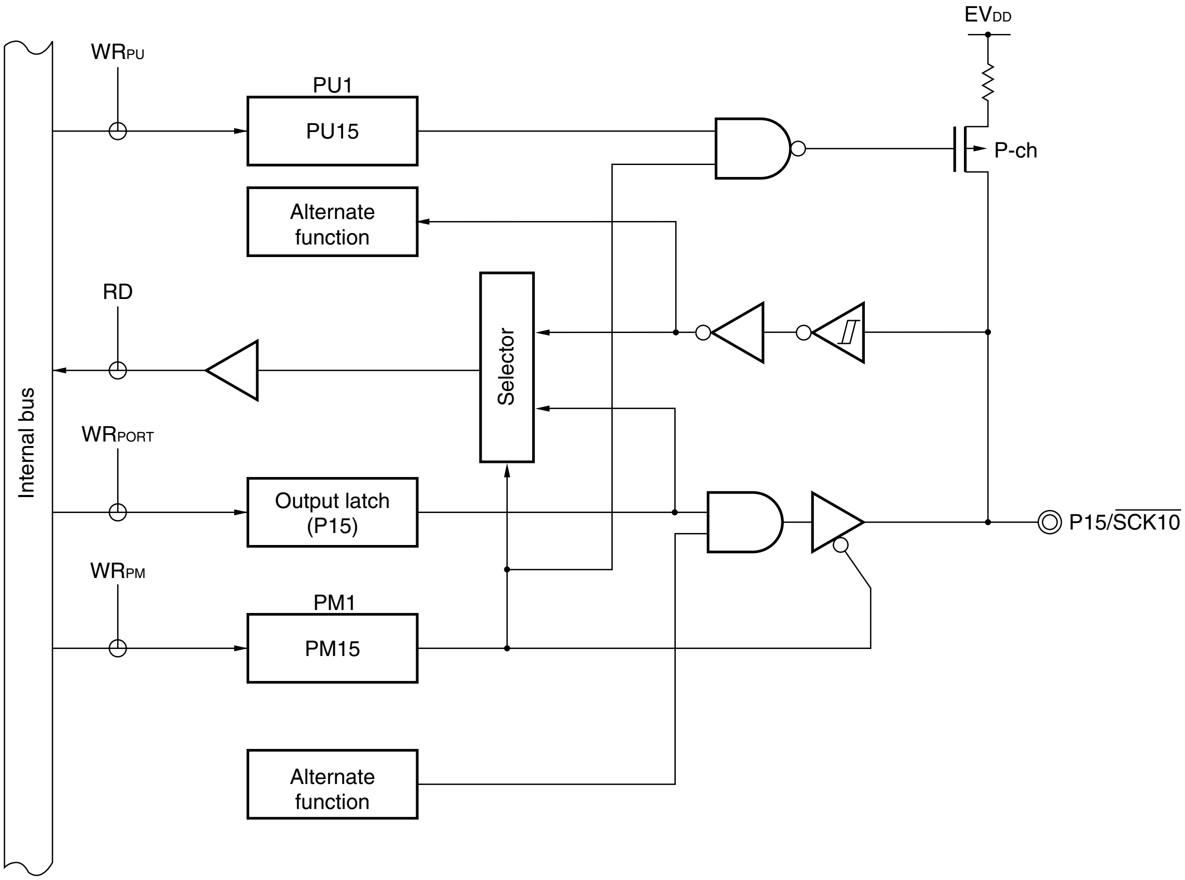
WR××: Write signal

Figure 4-4. Block Diagram of P14



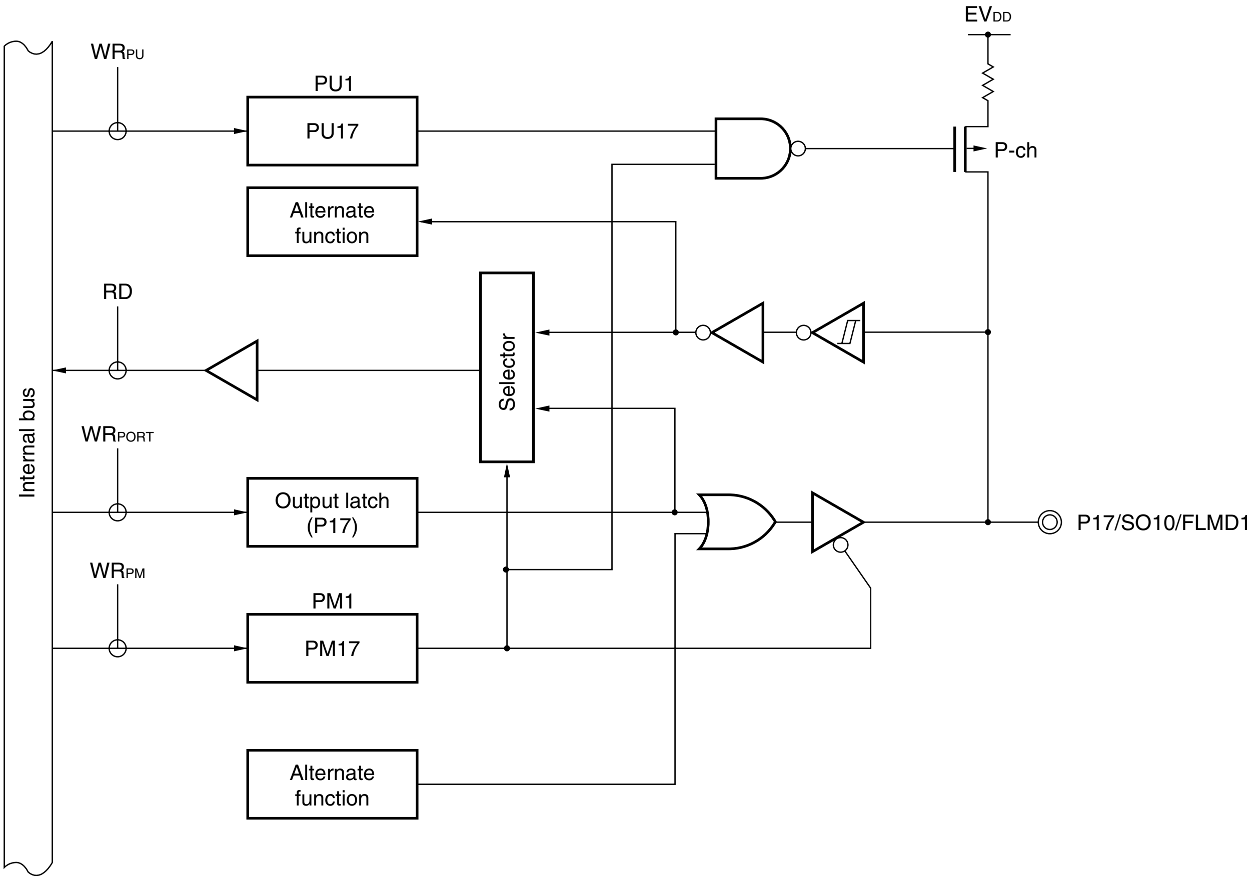
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-5. Block Diagram of P15



PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-6. Block Diagram of P17



PU1: Pull-up resistor option register 1

PM1: Port mode register 1

RD: Read signal

WR××: Write signal

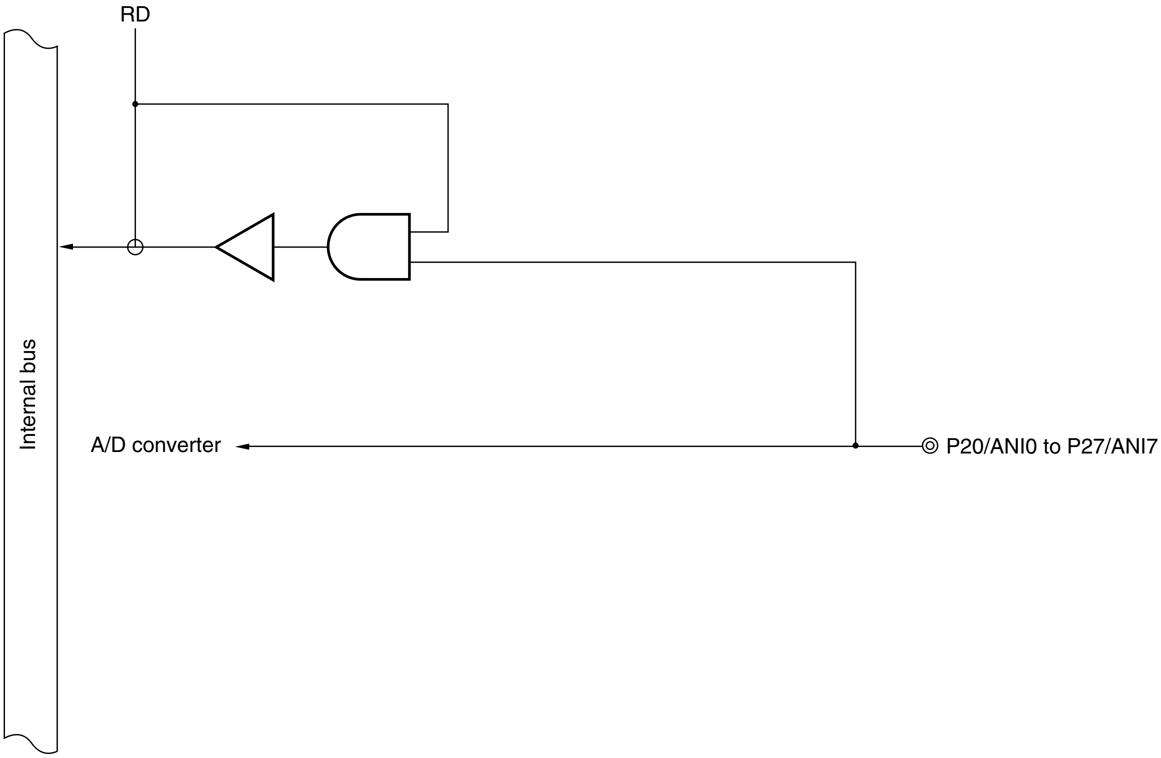
4.2.3 Port 2

Port 2 is an 8-bit input-only port.

This port can also be used for A/D converter analog input.

Figure 4-7 shows a block diagram of port 2.

Figure 4-7. Block Diagram of P20 to P27



RD: Read signal

<R> **Caution Use P20 to P27 at EV$_{DD}$ = AV$_{REF}$ when using them in the port mode.**

4.2.4 Port 3

Port 3 is a 4-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

This port can also be used for buzzer output, and clock output.

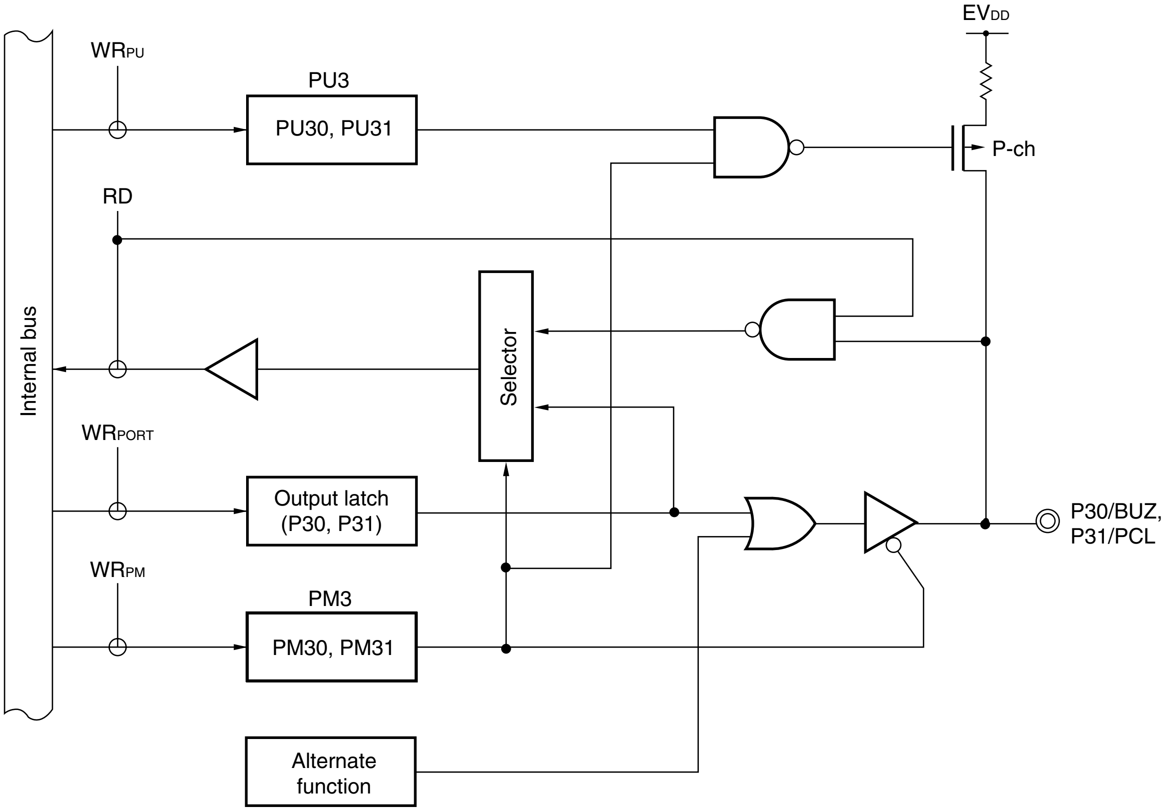
$\overline{\text{RESET}}$ input sets port 3 to input mode.

Figures 4-8 and 4-9 show block diagrams of port 3.

Caution Be sure to pull down P31 after reset to prevent malfunction.

Remark The P31/INTP2 and P32/INTP3 pins of the μPD78F0714 can be used to set the on-chip debug mode when the on-chip debug function is used. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

Figure 4-8. Block Diagram of P30 and P31



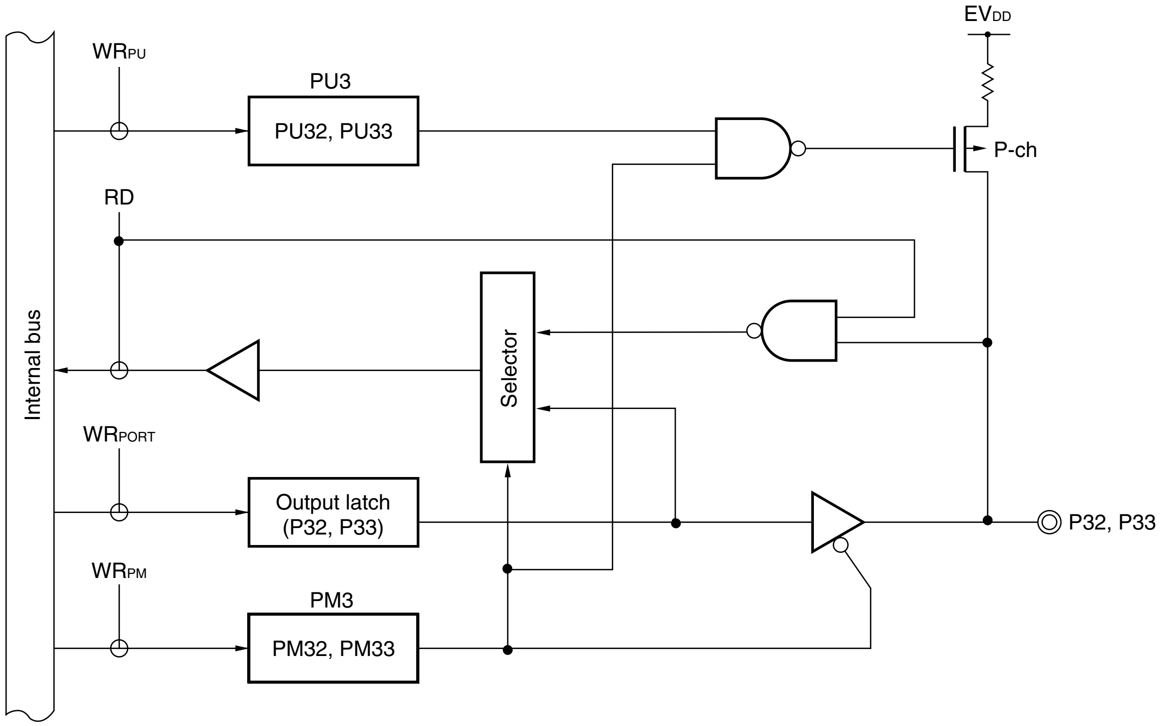
PU3: Pull-up resistor option register 3

PM3: Port mode register 3

RD: Read signal

WR××: Write signal

Figure 4-9. Block Diagram of P32 and P33



PU3: Pull-up resistor option register 3

PM3: Port mode register 3

RD: Read signal

WR××: Write signal

4.2.5 Port 4

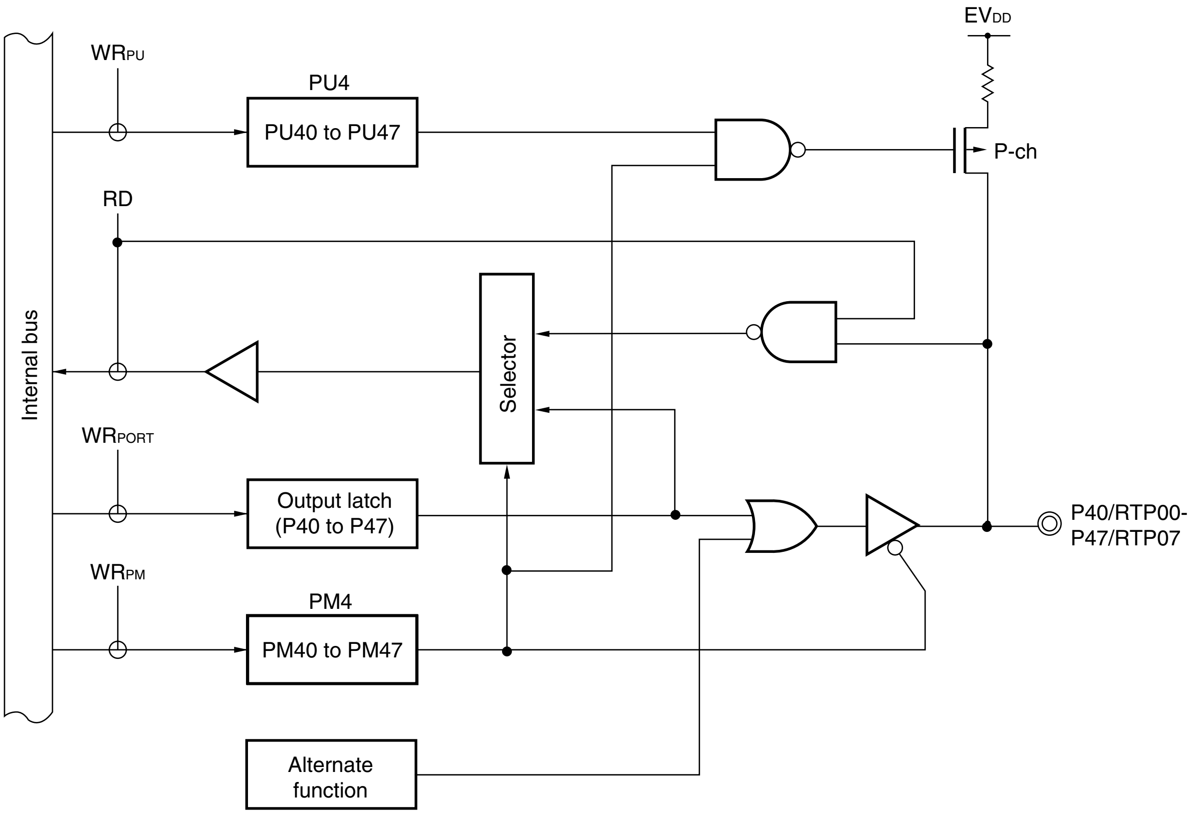
Port 4 is an 8-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified in 1-bit units with pull-up resistor option register 4 (PU4).

This port can also be used as real-time output ports.

$\overline{\text{RESET}}$ input sets port 4 to input mode.

Figure 4-10 shows a block diagram of port 4.

Figure 4-10. Block Diagram of P40 to P47



PU4: Pull-up resistor option register 4

PM4: Port mode register 4

RD: Read signal

WR××: Write signal

4.2.6 Port 5

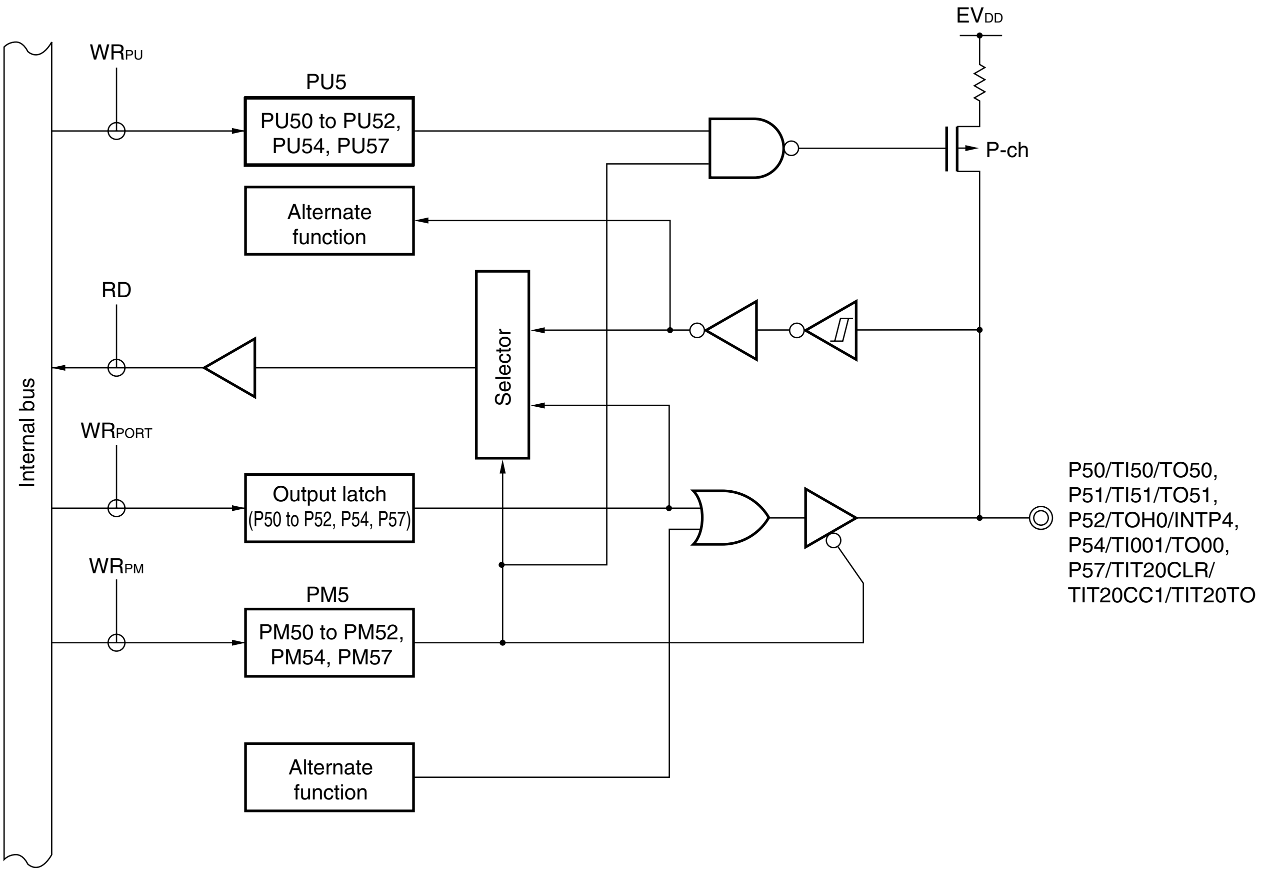
Port 5 is an 8-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified in 1-bit units using pull-up resistor option register 5 (PU5).

This port can also be used as external interrupt request input, timer I/O.

$\overline{\text{RESET}}$ input sets port 5 to input mode.

Figures 4-11 and 4-12 show block diagrams of port 5.

Figure 4-11. Block Diagram of P50 to P52, P54, and P57



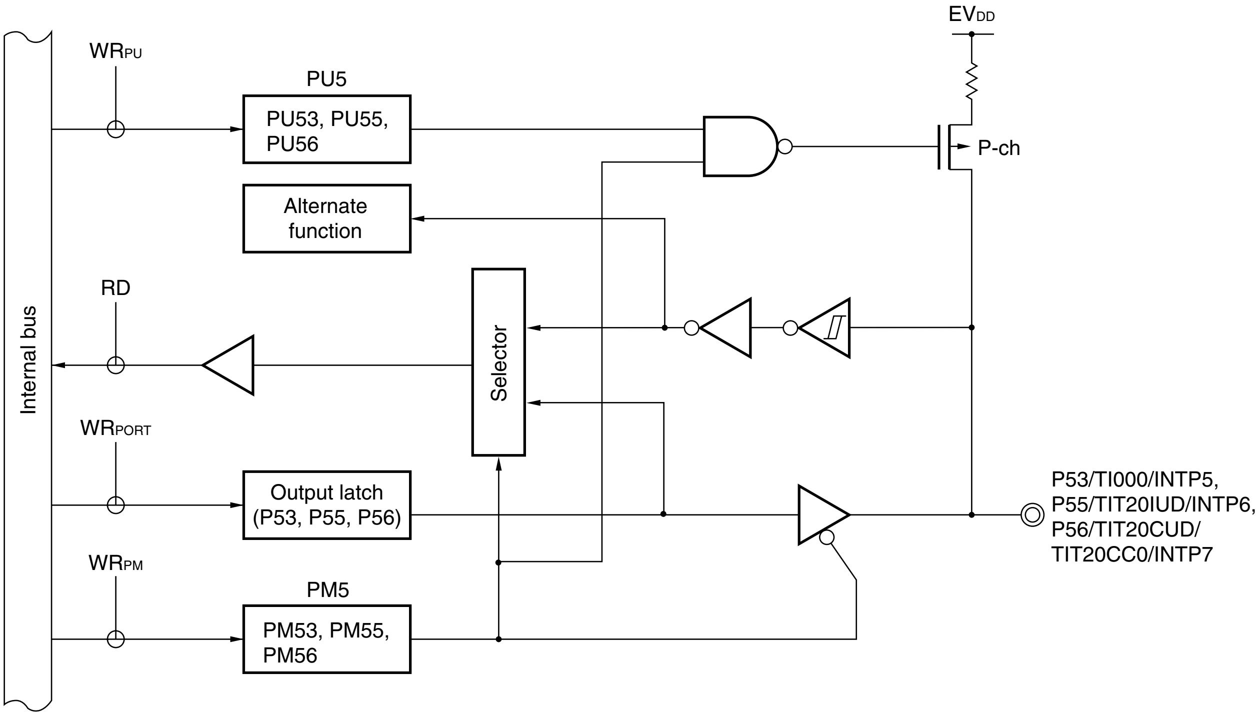
PU5: Pull-up resistor option register 5

PM5: Port mode register 5

RD: Read signal

WR××: Write signal

Figure 4-12. Block Diagram of P53, P55, and P56



PU5: Pull-up resistor option register 5

PM5: Port mode register 5

RD: Read signal

WR××: Write signal

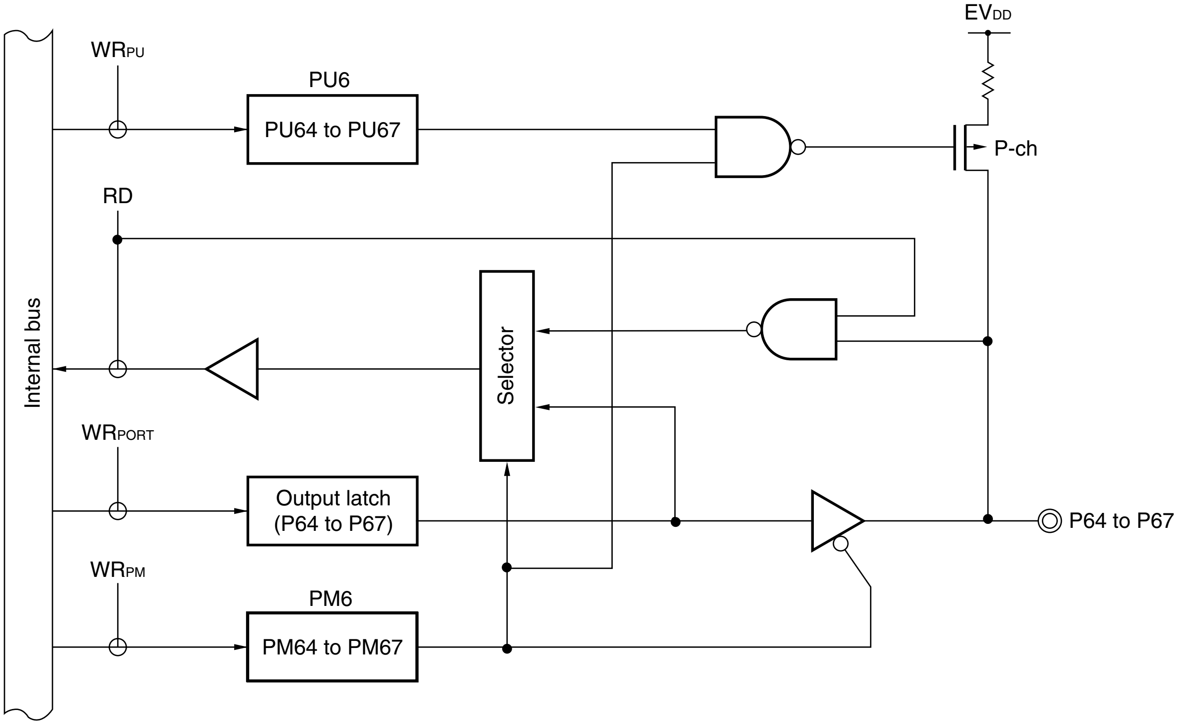
4.2.7 Port 6

Port 6 is a 4-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 6 (PU6).

$\overline{\text{RESET}}$ input sets port 6 to input mode.

Figure 4-13 shows a block diagram of port 6.

Figure 4-13. Block Diagram of P64 to P67



PU6: Pull-up resistor option register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

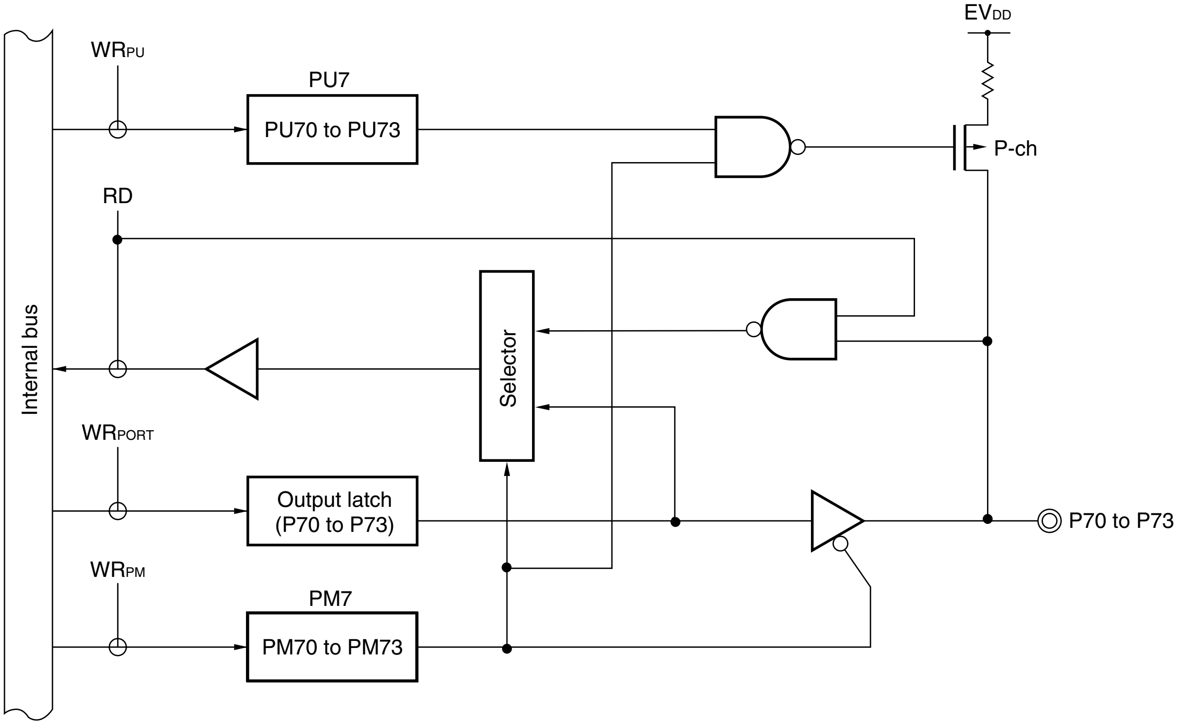
4.2.8 Port 7

Port 7 is an 4-bit I/O port with an output latch. Port 7 can be set to the input mode or output mode in 1-bit units using port mode register 7 (PM7). When the P70 to P73 pins are used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 7 (PU7).

$\overline{\text{RESET}}$ input sets port 7 to input mode.

Figure 4-14 shows a block diagram of port 7.

Figure 4-14. Block Diagram of P70 to P73



PU7: Pull-up resistor option register 7
PM7: Port mode register 7
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following three types of registers.

- Port mode registers (PM0, PM1, PM3 to PM7)
- Port registers (P0 to P7)
- Pull-up resistor option registers (PU0, PU1, PU3 to PU7)

(1) Port mode registers (PM0, PM1, PM3 to PM7)

These registers specify input or output mode for the port in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to FFH.

When port pins are used as alternate-function pins, set the port mode register and output latch as shown in Table 4-4.

Figure 4-15. Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	PM03	PM02	PM01	PM00	FF20H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FF21H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FF23H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FF25H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM6	PM67	PM66	PM65	PM64	1	1	1	1	FF26H	FFH	R/W
	7	6	5	4	3	2	1	0			
PM7	1	1	1	1	PM73	PM72	PM71	PM70	FF27H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 1, 3 to 7; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Table 4-4. Settings of Port Mode Register and Output Latch When Using Alternate Function

Pin Name	Alternate Function		PM××	P××
	Function Name	I/O		
P00	INTP0	Input	1	×
	TW0TOFFP	Input	1	×
P01	INTP1	Input	1	×
P02	INTP2	Input	1	×
P03	INTP3	Input	1	×
	ADTRG	Input	1	×
P13	RxD00	Input	1	×
P14	TxD00	Output	0	1
P15	$\overline{\text{SCK10}}$	Input	1	×
		Output	0	1
P16	SI10	Input	1	×
P17	SO10	Output	0	0
	FLMD1	Input	1	×
P20-P27	ANI0-ANI7	Input	1	×
P30	BUZ	Output	0	0
P31	PCL	Output	0	0
P40-P47	RTP00-RTP07	Output	0	0
P50	TI50	Input	1	×
	TO50	Output	0	0
P51	TI51	Input	1	×
	TO51	Output	0	0
P52	INTP4	Input	1	×
	TOH0	Output	0	0
P53	INTP5	Input	1	×
	TI000	Input	1	×
P54	TI001	Input	1	×
	TO00	Output	0	0
P55	INTP6	Input	1	×
	TIT20IUD	Input	1	×
P56	INTP7	Input	1	×
	TIT20CUD	Input	1	×
	TIT20CC0	Input	1	×
P57	TIT20CC1	Input	1	×
	TIT20CLR	Input	1	×
	TIT20TO	Output	0	0

Remark ×: Don't care

PM××: Port mode register

P××: Port output latch

(2) Port registers (P0 to P7)

These registers write the data that is output from the chip when data is output from a port.

If the data is read in the input mode, the pin level is read. If it is read in the output mode, the value of the output latch is read.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H (but P2 is undefined).

Figure 4-16. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	P03	P02	P01	P00	FF00H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P1	P17	P16	P15	P14	P13	P12	P11	P10	FF01H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P2	P27	P26	P25	P24	P23	P22	P21	P20	FF02H	Undefined	R
	7	6	5	4	3	2	1	0			
P3	0	0	0	0	P33	P32	P31	P30	FF03H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P4	P47	P46	P45	P44	P43	P42	P41	P40	FF04H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P5	P57	P56	P55	P54	P53	P52	P51	P50	FF05H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P6	P67	P66	P65	P64	0	0	0	0	FF06H	00H (output latch)	R/W
	7	6	5	4	3	2	1	0			
P7	0	0	0	0	P73	P72	P71	P70	FF07H	00H (output latch)	R/W

Pmn	m = 0 to 7; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

(3) Pull-up resistor option registers (PU0, PU1, and PU3 to PU7)

These registers specify whether the on-chip pull-up resistors of P00 to P03, P10 to P17, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P73 are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode of the pins to which the use of an on-chip pull-up resistor has been specified. On-chip pull-up resistors cannot be connected for bits set to output mode and bits used as alternate-function output pins, regardless of the settings of PU0, PU1, and PU3 to PU7.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 4-17. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	PU03	PU02	PU01	PU00	FF30H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	FF31H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	FF33H	00H	R/W
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40	FF34H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	FF35H	00H	R/W
PU6	PU67	PU66	PU65	PU64	0	0	0	0	FF36H	00H	R/W
PU7	0	0	0	0	PU73	PU72	PU71	PU70	FF37H	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 7; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

Caution In the case of a 1-bit memory manipulation instruction, although a single bit is manipulated, the port is accessed as an 8-bit unit. Therefore, on a port with a mixture of input and output pins, the output latch contents for pins specified as input are undefined, even for bits other than the manipulated bit.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared by reset.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.
Once data is written to the output latch, it is retained until data is written to the output latch again.

4.4.2 Reading from I/O port

(1) Output mode

The output latch contents are read by a transfer instruction. The output latch contents do not change.

(2) Input mode

The pin status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.
Once data is written to the output latch, it is retained until data is written to the output latch again.
The data of the output latch is cleared by reset.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change.

<R>

4.5 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the μPD78F0714.

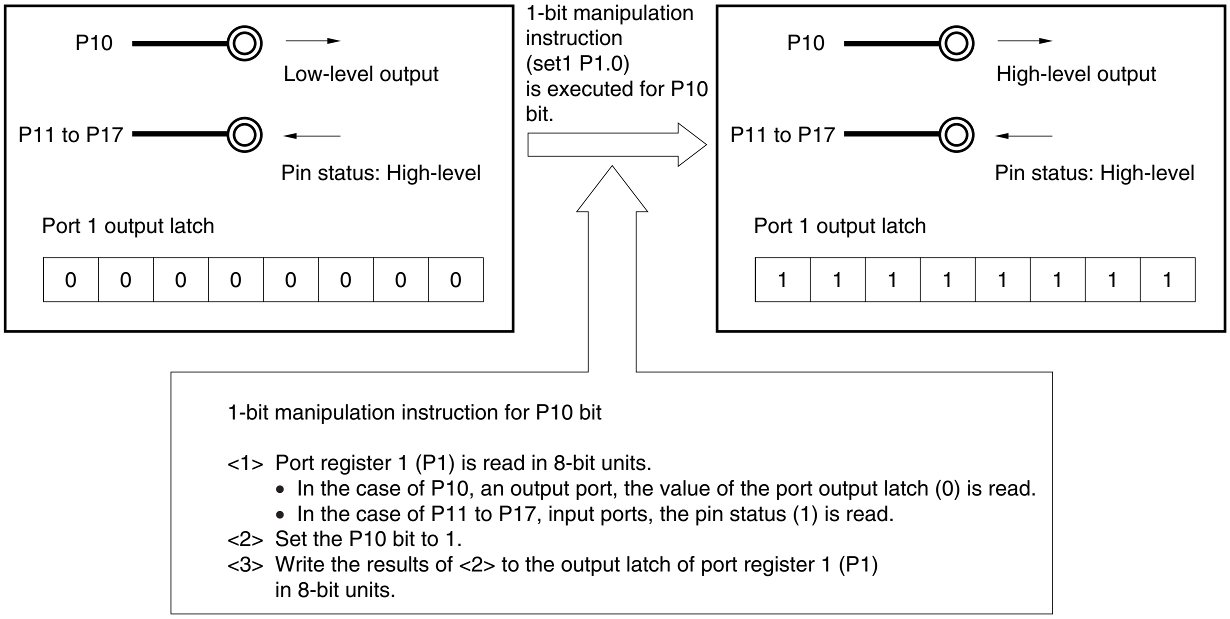
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4-18. Bit Manipulation Instruction (P10)



0	0	0	0	0	0	0	0

1	1	1	1	1	1	1	1

1-bit manipulation instruction for P10 bit

<1> Port register 1 (P1) is read in 8-bit units.
- In the case of P10, an output port, the value of the port output latch (0) is read.
- In the case of P11 to P17, input ports, the pin status (1) is read.

<2> Set the P10 bit to 1.
<3> Write the results of <2> to the output latch of port register 1 (P1) in 8-bit units.

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following two system clock oscillators are available.

- X1 oscillator
  The X1 oscillator oscillates a clock of f_{XP} = 5.0 to 20.0 MHz. Oscillation can be stopped by executing the STOP instruction or setting the main OSC control register (MOC) and processor clock control register (PCC).
- Internal oscillator
  The Internal oscillator oscillates a clock of f_R = 240 kHz (TYP.). Oscillation can be stopped by setting the internal oscillation mode register (RCM) when “Can be stopped by software” is set by an option byte and the X1 input clock is used as the CPU clock.

Remarks 1. f_{XP}: X1 input clock oscillation frequency
2. f_R: internal oscillation clock frequency

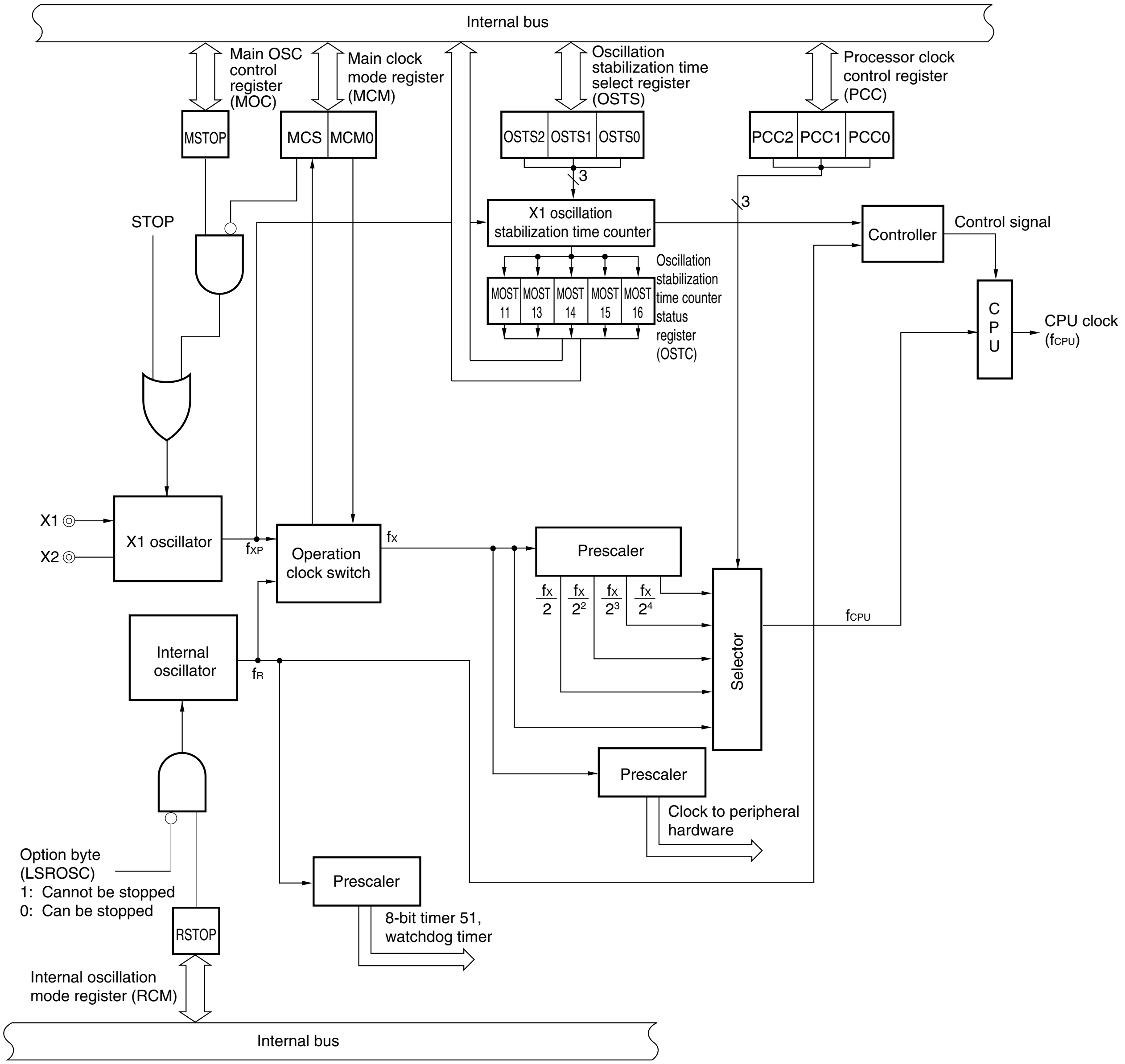
5.2 Configuration of Clock Generator

The clock generator consists of the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Processor clock control register (PCC) Internal oscillation mode register (RCM) Main clock mode register (MCM) Main OSC control register (MOC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System wait control register (VSWC)
Oscillator	X1 oscillator Internal oscillator

Figure 5-1. Block Diagram of Clock Generator


Internal bus
Main OSC control register (MOC)
Main clock mode register (MCM)
Oscillation stabilization time select register (OSTS)
Processor clock control register (PCC)
MSTOP
MCS
MCM0
OSTS2
OSTS1
OSTS0
PCC2
PCC1
PCC0
3
3
STOP
X1 oscillation stabilization time counter
Controller
Control signal
Oscillation stabilization time counter status register (OSTC)
MOST 11
MOST 13
MOST 14
MOST 15
MOST 16
C P U
CPU clock (fCPU)
X1
X2
X1 oscillator
fXP
Operation clock switch
fX
Prescaler
fX/2
fX/2²
fX/2³
fX/2⁴
Selector
fCPU
Internal oscillator
fR
Prescaler
Clock to peripheral hardware
Prescaler
8-bit timer 51, watchdog timer
Option byte (LSROSC)
1: Cannot be stopped
0: Can be stopped
RSTOP
Internal oscillation mode register (RCM)
Internal bus

5.3 Registers Controlling Clock Generator

The following seven registers are used to control the clock generator.

- Processor clock control register (PCC)
- Internal oscillation mode register (RCM)
- Main clock mode register (MCM)
- Main OSC control register (MOC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System wait control register (VSWC)

(1) Processor clock control register (PCC)

The PCC register is used to set the CPU clock division ratio.
The PCC is set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears PCC to 00H.

Figure 5-2. Format of Processor Clock Control Register (PCC)

Address: FFFBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	PCC2	PCC1	PCC0

PCC2	PCC1	PCC0	CPU clock (f_{CPU}) selection		
				MCM0 = 0	MCM0 = 1
0	0	0	f_X	f_R	f_{XP}
0	0	1	$f_X/2$	$f_R/2$	$f_{XP}/2$
0	1	0	$f_X/2^2$	–[Note]	$f_{XP}/2^2$
0	1	1	$f_X/2^3$	–[Note]	$f_{XP}/2^3$
1	0	0	$f_X/2^4$	–[Note]	$f_{XP}/2^4$
Other than above			Setting prohibited		

Note Setting prohibited.

Caution Be sure to clear bit 3 to 7 to 0.

Remarks 1. MCM0: Bit 0 of main clock mode register (MCM)

2. f_X: Main system clock oscillation frequency (X1 input clock oscillation frequency or internal oscillation clock frequency)

3. f_R: Internal oscillation clock frequency

4. f_{XP}: X1 input clock oscillation frequency

The fastest instruction can be executed in 2 clocks of the CPU clock in the μPD78F0714. Therefore, the relationship between the CPU clock (f_{CPU}) and minimum instruction execution time is as shown in the Table 5-2.

Table 5-2. Relationship Between CPU Clock and Minimum Instruction Execution Time

CPU Clock (f_{CPU})	Minimum Instruction Execution Time: $2/f_{CPU}$		
	X1 Input Clock[Note 1]		Internal Oscillation Clock[Note 1]
	At 20 MHz Operation	At 16 MHz Operation	At 240 kHz (TYP.) Operation
f_X	0.1 μs	0.125 μs	8.3 μs (TYP.)
$f_X/2$	0.2 μs	0.25 μs	16.6 μs (TYP.)
$f_X/2^2$	0.4 μs	0.5 μs	–[Note 2]
$f_X/2^3$	0.8 μs	1.0 μs	–[Note 2]
$f_X/2^4$	1.6 μs	2.0 μs	–[Note 2]

Notes 1. The main clock mode register (MCM) is used to set the CPU clock (X1 input clock/internal oscillation clock) (see **Figure 5-4**).

2. Setting prohibited.

(2) Internal oscillation mode register (RCM)

This register sets the operation mode of internal oscillator.

This register is valid when “Can be stopped by software” is set for internal oscillator by an option byte, and the X1 input clock is selected as the CPU clock. If “Cannot be stopped” is selected for internal oscillator by an option byte, settings for this register are invalid.

RCM can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 5-3. Format of Internal Oscillation Mode Register (RCM)

Address: FFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
RCM	0	0	0	0	0	0	0	RSTOP

RSTOP	Internal oscillator oscillating/stopped
0	Internal oscillator oscillating
1	Internal oscillator stopped

Caution Make sure that the bit 1 (MCS) of the main clock mode register (MCM) is 1 before setting RSTOP.

(3) Main clock mode register (MCM)

This register sets the CPU clock (X1 input clock/internal oscillation clock).

MCM can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 5-4. Format of Main Clock Mode Register (MCM)

Address: FFA1H After reset: 00H R/W[Note]

Symbol	7	6	5	4	3	2	<1>	<0>
MCM	0	0	0	0	0	0	MCS	MCM0

MCS	CPU clock status
0	Operates with internal oscillation clock
1	Operates with X1 input clock

MCM0	Selection of source clock to CPU
0	Internal oscillation clock
1	X1 input clock

Note Bit 1 is read-only.

Caution When internal oscillation clock is selected as the source clock to the CPU, the divided clock of the internal oscillator output (f_X) is supplied to the peripheral hardware (f_X = 240 kHz (TYP.)).

Operation of the peripheral hardware with internal oscillation clock cannot be guaranteed. Therefore, when internal oscillation clock is selected as the source clock to the CPU, do not use peripheral hardware. In addition, stop the peripheral hardware before switching the source clock to the CPU from the X1 input clock to the internal oscillation clock. Note, however, that the following peripheral hardware can be used when the CPU operates on the internal oscillation clock.

- **Watchdog timer**
- **8-bit timer 51 when $f_R/2^7$ is selected as count clock**
- **Peripheral hardware selecting external clock as the clock source (Except when external count clock of 16-bit up/down counter ITENC20 or 16-bit timer/event counter 00 is selected)**

(4) Main OSC control register (MOC)

This register selects the operation mode of the X1 input clock.

This register is used to stop the X1 oscillator operation when the CPU is operating with the internal oscillation clock. Therefore, this register is valid only when the CPU is operating with the internal oscillation clock.

MOC can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 5-5. Format of Main OSC Control Register (MOC)

Address: FFA2H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
MOC	MSTOP	0	0	0	0	0	0	0

MSTOP	Control of X1 oscillator operation
0	X1 oscillator operating
1	X1 oscillator stopped

Caution Make sure that bit 1 (MCS) of the main clock mode register (MCM) is 0 before setting MSTOP.

(5) Oscillation stabilization time counter status register (OSTC)

This is the status register of the X1 input clock oscillation stabilization time counter. If the internal oscillation clock is used as the CPU clock, the X1 input clock oscillation stabilization time can be checked.

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, and WDT), the STOP instruction, MSTOP = 1 clear OSTC to 00H.

Figure 5-6. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status	
						f_{XP} = 20 MHz
1	0	0	0	0	$2^{11}/f_{XP}$ min.	102.4 μs min.
1	1	0	0	0	$2^{13}/f_{XP}$ min.	409.6 μs min.
1	1	1	0	0	$2^{14}/f_{XP}$ min.	819.2 μs min.
1	1	1	1	0	$2^{15}/f_{XP}$ min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_{XP}$ min.	3.27 ms min.

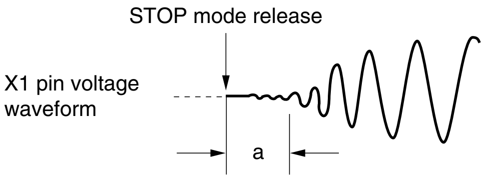
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the internal oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The X1 oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

3. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts ("a" below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_{XP}: X1 input clock oscillation frequency

(6) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 oscillation stabilization wait time when STOP mode is released.

The wait time set by OSTS is valid only after STOP mode is released with the X1 input clock selected as CPU clock. After STOP mode is released with internal oscillation clock selected as CPU clock, the oscillation stabilization time must be confirmed by OSTC.

OSTS can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 05H.

Figure 5-7. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

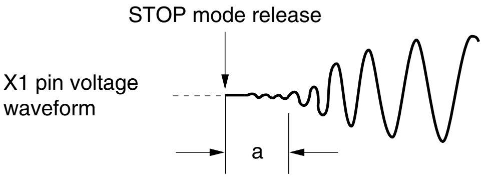
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	
				f_{XP} = 20 MHz
0	0	1	$2^{11}/f_{XP}$	102.4 μs
0	1	0	$2^{13}/f_{XP}$	409.6 μs
0	1	1	$2^{14}/f_{XP}$	819.2 μs
1	0	0	$2^{15}/f_{XP}$	1.64 ms
1	0	1	$2^{16}/f_{XP}$	3.27 ms
Other than above			Setting prohibited	

Cautions 1. If the STOP mode is entered and then released while the internal oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The X1 oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

2. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts (“a” below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_{XP}: X1 input clock oscillation frequency

(7) System wait control register (VSWC)

This register is used to control wait states when a high-speed CPU and a low-speed peripheral I/O are connected.

VSWC can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 5-8. Format of System Wait Control Register (VSWC)

Address: FFFDH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
VSWC	0	0	0	0	0	0	PDW1	0

PDW1	Control of system clock data wait
0	No wait
1	Two wait states inserted

Cautions 1. Be sure to insert two wait states if the minimum instruction execution time is 0.125 μs or less (f_{XP} = 16 MHz or more).

2. Be sure to clear bits 0 and 2 to 7 to 0.

5.4 System Clock Oscillator

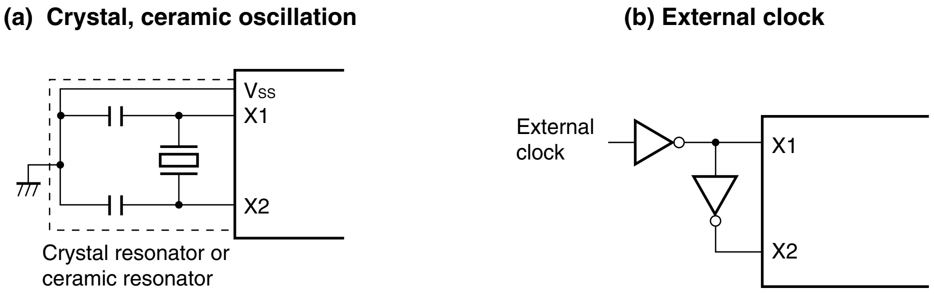
5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (Standard: 20 MHz) connected to the X1 and X2 pins.

An external clock can be input to the X1 oscillator. In this case, input the clock signal to the X1 pin and input the inverse signal to the X2 pin.

Figure 5-9 shows examples of the external circuit of the X1 oscillator.

Figure 5-9. Examples of External Circuit of X1 Oscillator



(a) Crystal, ceramic oscillation

(b) External clock

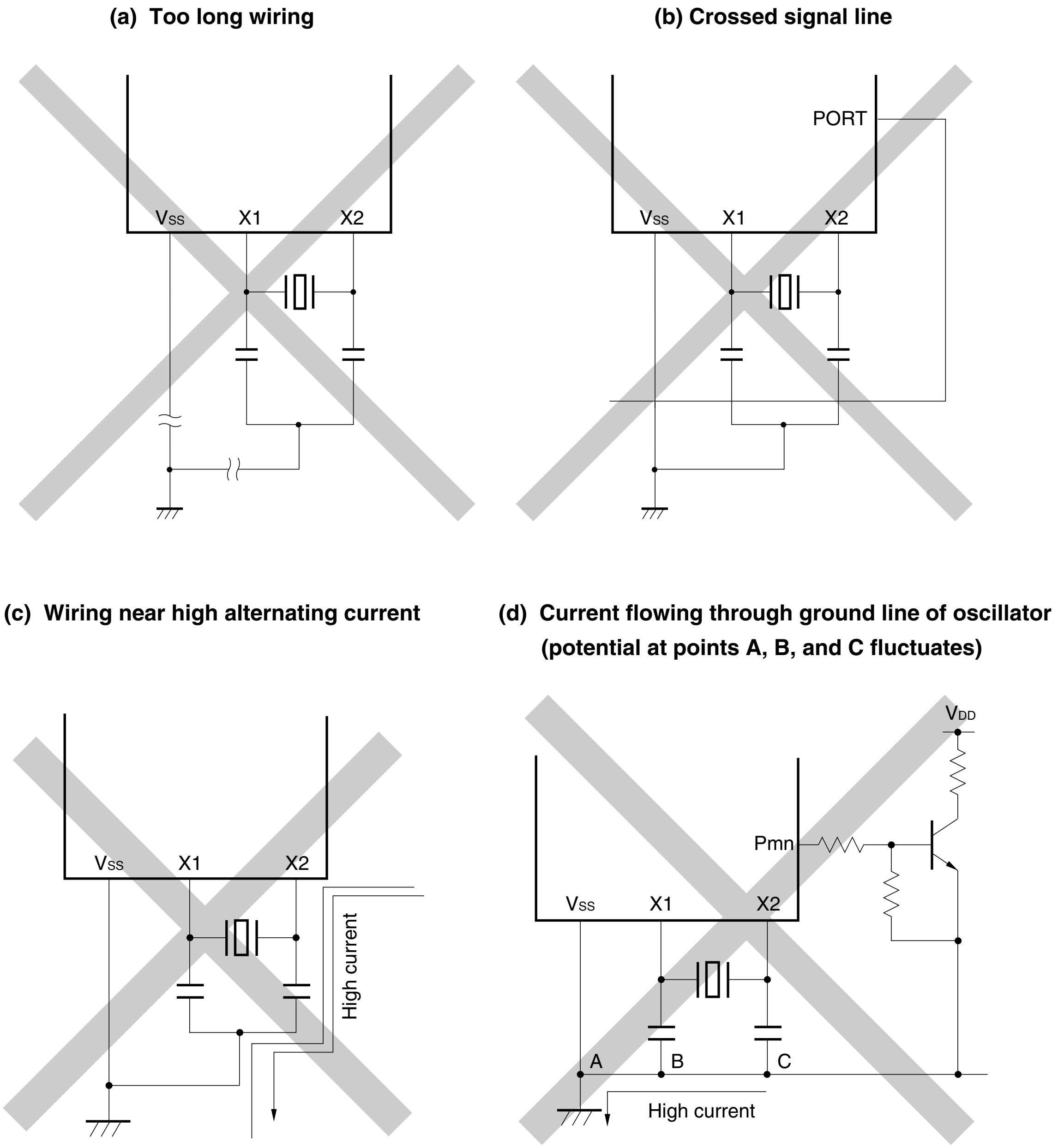
Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figure 5-9 to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as VSS. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

5.4.2 Examples of Incorrect Resonator Connection

Figure 5-10 shows examples of incorrect resonator connection.

Figure 5-10. Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

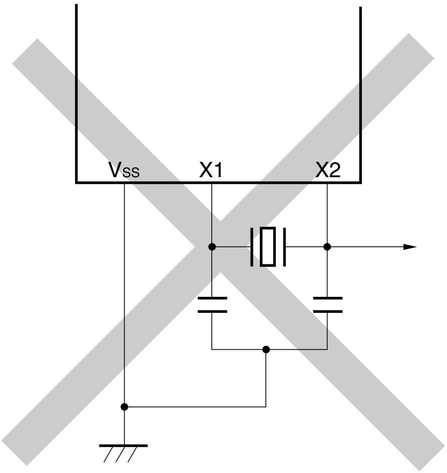
(b) Crossed signal line

(c) Wiring near high alternating current

(d) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)

Figure 5-10. Examples of Incorrect Resonator Connection (2/2)

(e) Signals are fetched



5.4.3 Internal oscillator

Internal oscillator is incorporated in the μPD78F0714.

“Can be stopped by software” or “Cannot be stopped” can be selected by an option byte. The internal oscillation clock always oscillates after $\overline{\text{RESET}}$ release (240 kHz (TYP.)).

5.4.4 Prescaler

The prescaler generates various clocks by dividing the X1 oscillator output when the X1 input clock is selected as the source clock to the CPU.

Caution When the internal oscillation clock is selected as the source clock to the CPU, the prescaler generates various clocks by dividing the internal oscillator output (f_X = 240 kHz (TYP.)).

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode.

- X1 input clock f_{XP}
- Internal oscillation clock f_R
- CPU clock f_{CPU}
- Clock to peripheral hardware

The CPU starts operation when the on-chip internal oscillator starts outputting after reset release in the μPD78F0714, thus enabling the following.

(1) Enhancement of security function

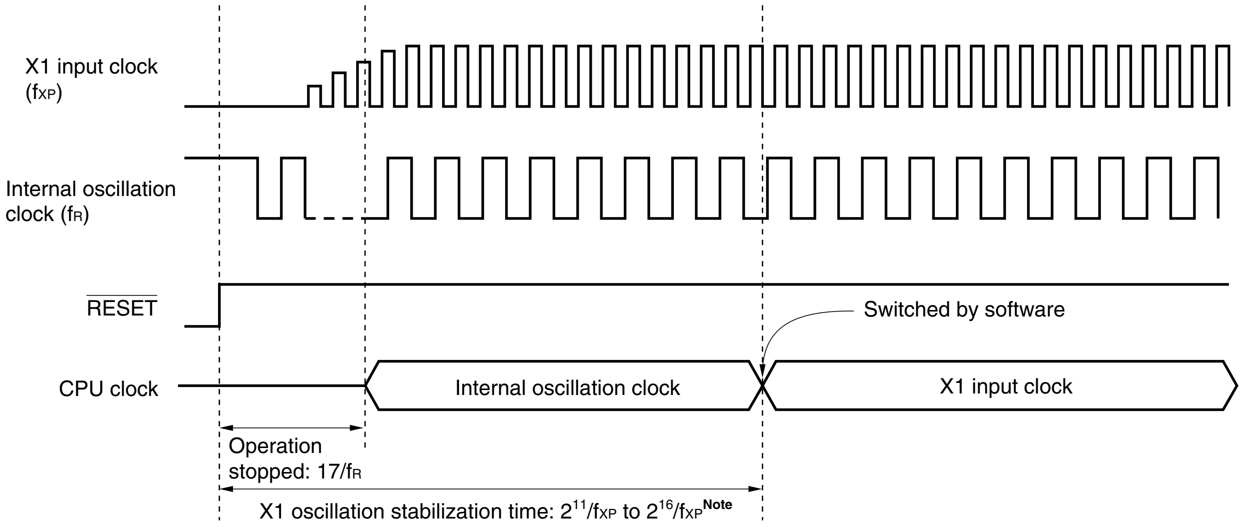
When the X1 input clock is set as the CPU clock by the default setting, the device cannot operate if the X1 input clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the on-chip internal oscillation clock, so the device can be started by the internal oscillation clock after reset release. Consequently, the system can be safely shut down by performing a minimum operation, such as acknowledging a reset source by software or performing safety processing when there is a malfunction.

(2) Improvement of performance

Because the CPU can be started without waiting for the X1 input clock oscillation stabilization time, the total performance can be improved.

A timing diagram of the CPU default start using internal oscillator is shown in Figure 5-11.

Figure 5-11. Timing Diagram of CPU Default Start Using Internal Oscillator



Note Check using the oscillation stabilization time counter status register (OSTC).

(a) When the RESET signal is generated, bit 0 of the main clock mode register (MCM) is cleared to 0 and the internal oscillation clock is set as the CPU clock. However, a clock is supplied to the CPU after 17 clocks of the internal oscillation clock have elapsed after RESET release (or clock supply to the CPU stops for 17 clocks). During the RESET period, oscillation of the X1 input clock and internal oscillation clock is stopped.

(b) After RESET release, the CPU clock can be switched from the internal oscillation clock to the X1 input clock using bit 0 (MCM0) of the main clock mode register (MCM) after the X1 input clock oscillation stabilization time has elapsed. At this time, check the oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) before switching the CPU clock. The CPU clock status can be checked using bit 1 (MCS) of MCM.

(c) Internal oscillator can be set to stopped/oscillating using the internal oscillation mode register (RCM) when “Can be stopped by software” is selected for the internal oscillation clock by an option byte, if the X1 input clock is used as the CPU clock. Make sure that MCS is 1 at this time.

(d) When internal oscillation clock is used as the CPU clock, the X1 input clock can be set to stopped/oscillating using the main OSC control register (MOC). Make sure that MCS is 0 at this time.

(e) The oscillation stabilization time ($2^{11}/f_{XP}$, $2^{13}/f_{XP}$, $2^{14}/f_{XP}$, $2^{15}/f_{XP}$, $2^{16}/f_{XP}$) selected by the oscillation stabilization time select register (OSTS) is secured when releasing STOP mode while the X1 input clock is being used as the CPU clock.

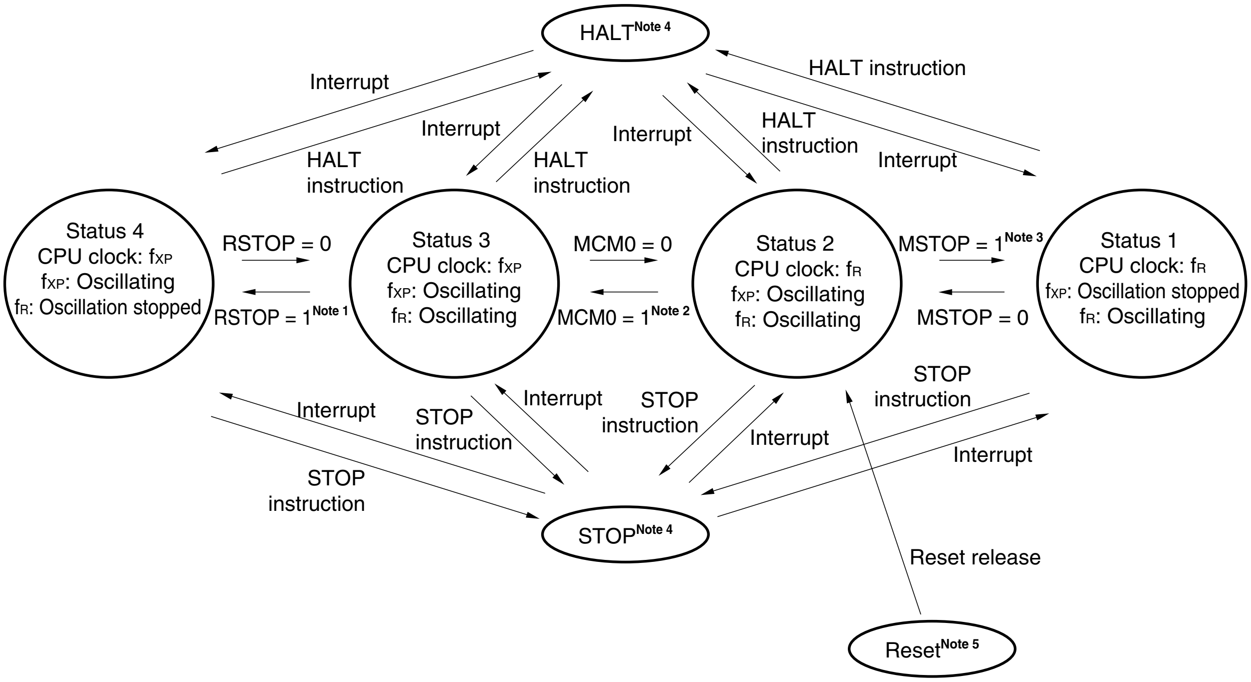
In addition, when RESET is released, and when the STOP mode is released while the internal oscillation clock is being used as the CPU clock, there is no oscillation stabilization time wait.

When switching to the X1 input clock as the CPU clock, check the oscillation stabilization time by using the oscillation stabilization time counter status register (OSTC).

A status transition diagram of this product is shown in Figure 5-12, and the relationship between the operation clocks in each operation status and between the oscillation control flag and oscillation status of each clock are shown in Tables 5-3 and 5-4, respectively.

Figure 5-12. Status Transition Diagram (1/2)

(1) When “Internal oscillator can be stopped by software” is selected by option byte



Notes 1. When shifting from status 3 to status 4, make sure that bit 1 (MCS) of the main clock mode register (MCM) is 1.

2. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

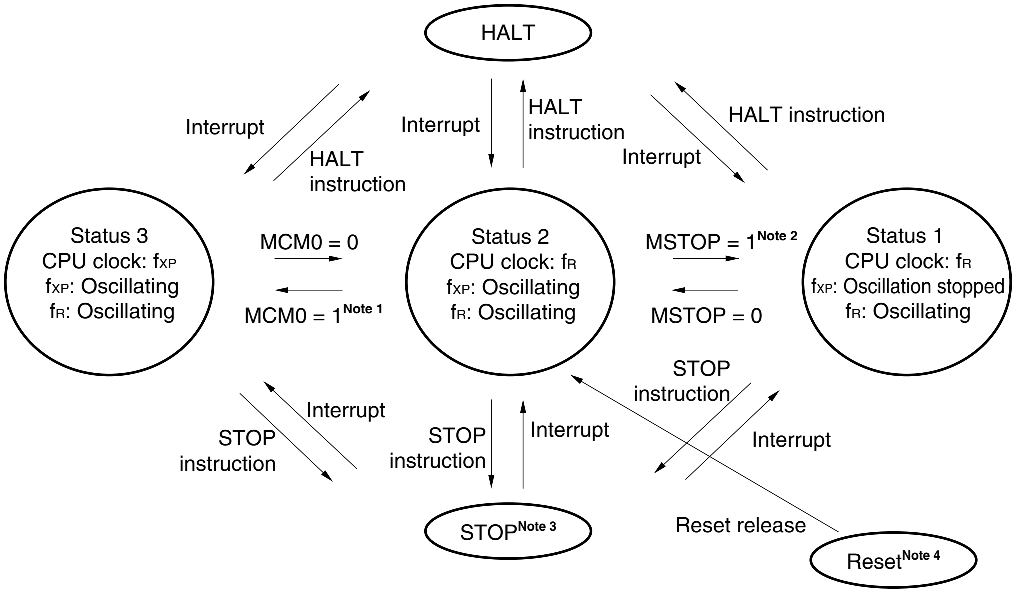
3. When shifting from status 2 to status 1, make sure that MCS is 0.

4. When “Internal oscillator can be stopped by software” is selected by an option byte, the watchdog timer stops operating in the HALT and STOP modes, regardless of the source clock of the watchdog timer. However, oscillation of internal oscillator does not stop even in the HALT and STOP modes if RSTOP = 0.

5. All reset sources ($\overline{\text{RESET}}$ input, POC, LVI, and WDT)

Figure 5-12. Status Transition Diagram (2/2)

(2) When “Internal oscillator cannot be stopped” is selected by option byte



Notes 1. Before shifting from status 2 to status 3 after reset and STOP are released, check the X1 input clock oscillation stabilization time status using the oscillation stabilization time counter status register (OSTC).

2. When shifting from status 2 to status 1, make sure that MCS is 0.

3. The watchdog timer operates using Internal oscillation clock even in STOP mode if “Internal oscillator cannot be stopped” is selected by an option byte. Internal oscillation clock division can be selected as the count source of 8-bit timer 51 (TM51), so clear the watchdog timer using the TM51 interrupt request before watchdog timer overflow. If this processing is not performed, an internal reset signal is generated at watchdog timer overflow after STOP instruction execution.

4. All reset sources ($\overline{\text{RESET}}$ input, POC, LVI, and WDT)

Table 5-3. Relationship Between Operation Clocks in Each Operation Status

Status Operation Mode	X1 Oscillator		Internal Oscillator			CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
	MSTOP = 0	MSTOP = 1	**Note 1**	**Note 2**				
				RSTOP = 0	RSTOP = 1		MCM0 = 0	MCM0 = 1
Reset	Stopped		Stopped			Internal oscillation clock	Stopped	
STOP			Oscillating	Oscillating	Stopped **Note 4**	**Note 5**	Stopped	
HALT	Oscillating	Stopped **Note 3**				**Note 6**	Internal oscillation clock	X1

Notes 1. When “Cannot be stopped” is selected for internal oscillator by an option byte.

2. When “Can be stopped by software” is selected for internal oscillator by an option byte.

3. Only when internal oscillator is oscillating.

4. Only when X1 oscillator is oscillating.

5. Operates using the CPU clock at STOP instruction execution.

6. Operates using the CPU clock at HALT instruction execution.

Caution The RSTOP setting is valid only when “Can be stopped by software” is set for internal oscillator by an option byte.

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
RSTOP: Bit 0 of the internal oscillation mode register (RCM)
MCM0: Bit 0 of the main clock mode register (MCM)

Table 5-4. Oscillation Control Flags and Clock Oscillation Status

		X1 Oscillator	Internal Oscillator
MSTOP = 1	RSTOP = 0	Stopped	Oscillating
	RSTOP = 1	Setting prohibited	
MSTOP = 0	RSTOP = 0	Oscillating	Oscillating
	RSTOP = 1		Stopped

Caution The RSTOP setting is valid only when “Can be stopped by software” is set for internal oscillator by an option byte.

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
RSTOP: Bit 0 of the internal oscillation mode register (RCM)

5.6 Time Required to Switch Between Internal Oscillation Clock and X1 Input Clock

Bit 0 (MCM0) of the main clock mode register (MCM) is used to switch between the internal oscillation clock and X1 input clock.

In the actual switching operation, switching does not occur immediately after MCM0 rewrite; several instructions are executed using the pre-switch clock after switching MCM0 (see **Table 5-5**).

Bit 1 (MCS) of MCM is used to judge that operation is performed using either the internal oscillation clock or X1 input clock.

To stop the original clock after switching the clock, wait for the number of clocks shown in Table 5-5.

Table 5-5. Maximum Time Required to Switch Between Internal Oscillation Clock and X1 Input Clock

PCC			Time Required for Switching	
PCC2	PCC1	PCC0	X1→ Internal Oscillation Clock	Internal Oscillation Clock → X1
0	0	0	f_{XP}/f_R + 1 clock	2 clocks
0	0	1	$f_{XP}/2f_R$ + 1 clock	

Caution To calculate the maximum time, set f_R = 120 kHz.

Remarks 1. PCC: Processor clock control register

2. f_{XP}: X1 input clock oscillation frequency

3. f_R: Internal oscillation clock frequency

4. The maximum time is the number of clocks of the CPU clock before switching.

5.7 Time Required for CPU Clock Switchover

The CPU clock can be switched using bits 0 to 2 (PCC0 to PCC2) of the processor clock control register (PCC).

The actual switchover operation is not performed immediately after rewriting to the PCC; operation continues on the pre-switchover clock for several instructions (see **Table 5-6**).

Table 5-6. Maximum Time Required for CPU Clock Switchover

Set Value Before Switchover			Set Value After Switchover														
PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0	PCC2	PCC1	PCC0
			0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	0				16 clocks			16 clocks			16 clocks			16 clocks		
0	0	1	8 clocks						8 clocks			8 clocks			8 clocks		
0	1	0	4 clocks			4 clocks						4 clocks			4 clocks		
0	1	1	2 clocks			2 clocks			2 clocks						2 clocks		
1	0	0	1 clock			1 clock			1 clock			1 clock					

Caution Setting the following values is prohibited when the CPU operates on the internal oscillation clock.

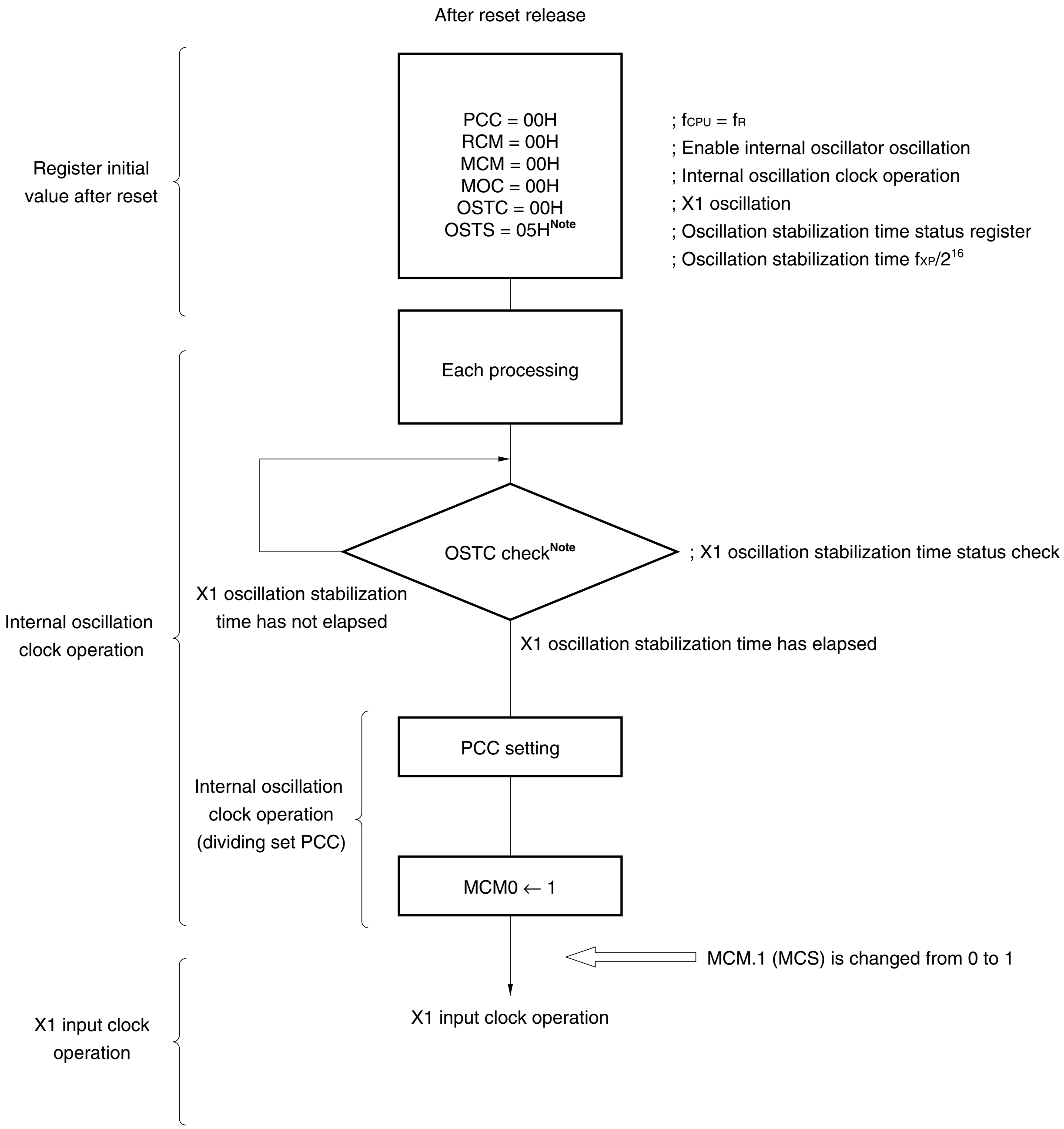
- **PCC2, PCC1, PCC0 = 0, 1, 0**
- **PCC2, PCC1, PCC0 = 0, 1, 1**
- **PCC2, PCC1, PCC0 = 1, 0, 0**

Remark The maximum time is the number of clocks of the CPU clock before switching.

5.8 Clock Switching Flowchart and Register Setting

5.8.1 Switching from internal oscillation clock to X1 input clock

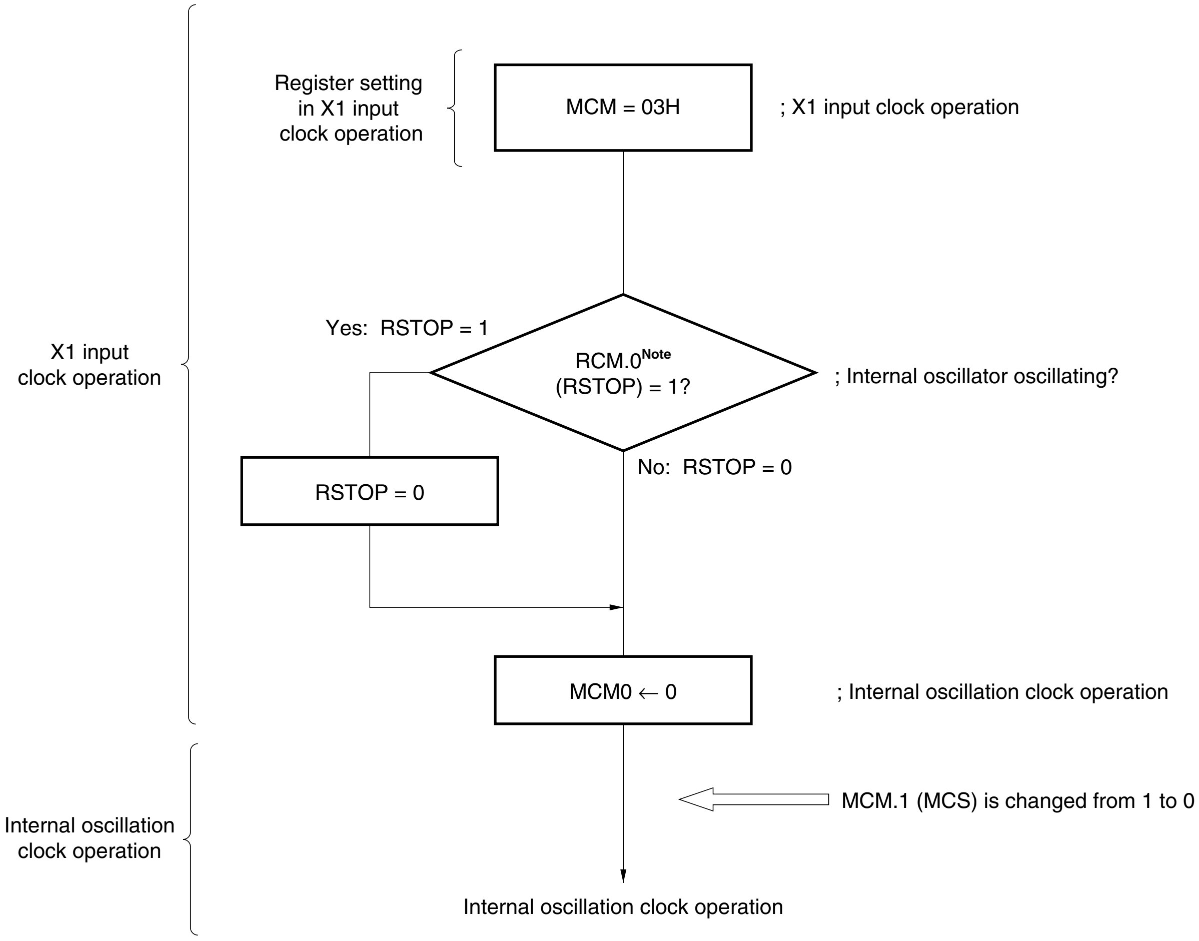
Figure 5-13 Switching from Internal Oscillation Clock to X1 Input Clock (Flowchart)


After reset release
Register initial value after reset
PCC = 00H
RCM = 00H
MCM = 00H
MOC = 00H
OSTC = 00H
OSTS = 05H[Note]
; $f_{CPU} = f_R$
; Enable internal oscillator oscillation
; Internal oscillation clock operation
; X1 oscillation
; Oscillation stabilization time status register
; Oscillation stabilization time $f_{XP}/2^{16}$
Each processing
Internal oscillation clock operation
OSTC check[Note]
; X1 oscillation stabilization time status check
X1 oscillation stabilization time has not elapsed
X1 oscillation stabilization time has elapsed
PCC setting
Internal oscillation clock operation (dividing set PCC)
MCM0 ← 1
MCM.1 (MCS) is changed from 0 to 1
X1 input clock operation
X1 input clock operation

Note Check the oscillation stabilization wait time of the X1 oscillator after reset release using the OSTC register and then switch to the X1 input clock operation after the oscillation stabilization wait time has elapsed. The OSTS register setting is valid only after STOP mode is released by interrupt during X1 input clock operation.

5.8.2 Switching from X1 input clock to internal oscillation clock

Figure 5-14 Switching from X1 Input Clock to Internal Oscillation Clock (Flowchart)


X1 input clock operation
Register setting in X1 input clock operation
MCM = 03H
; X1 input clock operation
Yes: RSTOP = 1
RCM.0Note (RSTOP) = 1?
; Internal oscillator oscillating?
RSTOP = 0
No: RSTOP = 0
MCM0 ← 0
; Internal oscillation clock operation
MCM.1 (MCS) is changed from 1 to 0
Internal oscillation clock operation
Internal oscillation clock operation

Note Required only when “clock can be stopped by software” is selected for internal oscillator by an option byte.

5.8.3 Register settings

The table below shows the statuses of the setting flags and status flags when each mode is set.

Table 5-7. Clock and Register Setting

f_{CPU}	Mode	Setting Flag			Status Flag
		MCM Register	MOC Register	RCM Register	MCM Register
		MCM0	MSTOP	RSTOP[Note 1]	MCS
X1 input clock[Note 2]	Internal oscillator oscillating	1	0	0	1
	Internal oscillator stopped	1	0	1	1
Internal oscillation clock	X1 oscillating	0	0	0	0
	X1 stopped	0	1	0	0

Notes 1. Valid only when "clock can be stopped by software" is selected for internal oscillator by an option byte.

2. Do not set MSTOP = 1 during X1 input clock operation (even if MSTOP = 1 is set, the X1 oscillation does not stop).

CHAPTER 6 10-BIT INVERTER CONTROL TIMER

6.1 Outline of 10-Bit Inverter Control Timer

The 10-bit inverter control timer makes inverter control possible. It consists of an 8-bit dead-time generation timer, and allows non-overlapping active-level output.

6.2 Function of 10-Bit Inverter Control Timer

The 10-bit inverter control timer realizes inverter control. It incorporates an 8-bit timer for dead time generation and can output waveforms that do not overlap active levels. A total of six positive phase and negative phase channels are output. In addition, an active level change function and output off function by external input (TW0TOFFP) are provided.

6.3 Configuration of 10-Bit Inverter Control Timer

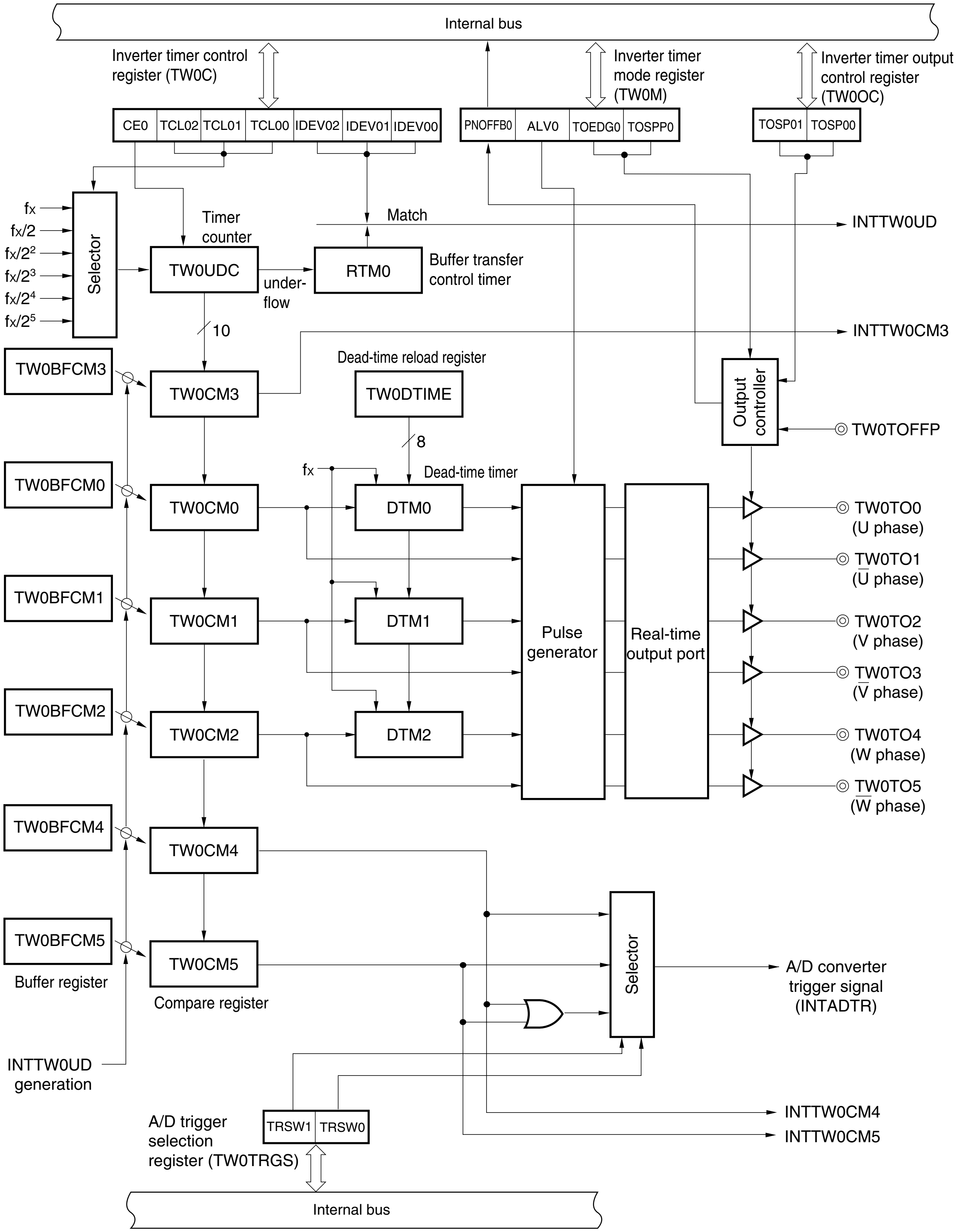
The 10-bit inverter control timer includes the following hardware.

Table 6-1. Configuration of 10-Bit Inverter Control Timer

Item	Function
Timer counter	10-bit up/down counter × 1 (TW0UDC) Dead-time timers × 3 (DTM0, DTM1, DTM2) Buffer transfer control timer × 1 (RTM0)
Register	10-bit compare registers × 6 (TW0CM0 to TW0CM5) 10-bit buffer registers × 6 (TW0BFCM0 to TW0BFCM5) Dead-time reload register × 1 (TW0DTIME)
Timer output	6 (TW0TO0, TW0TO1, TW0TO2, TW0TO3, TW0TO4, TW0TO5)
Control registers	Inverter timer control register (TW0C) Inverter timer mode register (TW0M) A/D trigger selection register (TW0TRGS) Inverter timer output control register (TW0OC)

Figure 6-1. Block Diagram of 10-Bit Inverter Control Timer

<R>



(1) 10-bit up/down counter (TW0UDC)

TW0UDC is a 10-bit up/down counter that counts count pulses in synchronization with the rising edge of the count clock. When the timer starts, the number of count pulse count is incremented from 0, and when the value preset to compare register 3 (TW0CM3) and TW0UDC count value match, it is switched to the count down operation.

An underflow signal is generated if the value becomes 000H during the count down operation and interrupt request signal INTTW0UD is generated. When an underflow occurs, it is switched from the count down operation to the count up operation. INTTW0UD is normally generated at every underflow but the number of occurrences can be divided by the IDEV00 to IDEV02 bits of inverter timer control register (TW0C).

TW0UDC cannot be read/written.

The cycle of TW0UDC is controlled by TM0CM3.

The count clock can be selected from 6 types: fx, fx/2, fx/4, fx/8, fx/16, fx/32.

$\overline{\text{RESET}}$ input or clearing the CE0 bit of TW0C7 sets TW0UDC to 000H.

(2) 10-bit compare registers 0 to 2 (TW0CM0 to TW0CM2)

TW0CM0 to TW0CM2 are 10-bit compare registers that always compare their own value with that of TW0UDC, and if they match, the contents of the flip-flops are changed.

Each of TW0CM0 to TW0CM2 are provided with a buffer register (TW0BFCM0 to TW0BFCM2), so that the contents of the buffer can be transferred to TW0CM0 to TW0CM2 at the timing of interrupt request signal INTTW0UD generation.

A write operation to TW0CM0 to TW0CM2 is possible only while TW0UDC is stopped.

To set the output timing, write data to TW0BFCM0 to TW0BFCM2.

$\overline{\text{RESET}}$ input or clearing the CE0 bit of TW0C sets these registers to 000H.

(3) 10-bit compare register 3 (TW0CM3)

TW0CM3 is a 10-bit compare register that controls the high limit value of TW0UDC. If the count value of TW0UDC matches the value of TW0CM3 or 0, count up/down is switched at the next count clock.

TW0CM3 provides a buffer register (TW0BFCM3) whose contents are transferred to TW0CM3 at the timing of interrupt request signal INTTW0UD generation.

TW0CM3 can be written to only while TW0UDC is stopped.

To set the cycle to TW0UDC, write data to TW0BFCM3.

$\overline{\text{RESET}}$ input sets TW0CM3 to 0FFH.

Do not set TW0CM3 to 000H.

(4) 10-bit compare registers 4, 5 (TW0CM4, TW0CM5)

TW0CM4 and TW0CM5 are 10-bit compare registers that always compare their own value with that of TW0UDC, and if they match, interrupt request signal is generated.

Each of TW0CM4 and TW0CM5 are provided with a buffer register (TW0BFCM4, TW0BFCM5), so that the contents of the buffer can be transferred to TW0CM4 to TW0CM5 at the timing of interrupt request signal INTTW0UD generation.

A write operation to TW0CM4 and TW0CM5 is possible only while TW0UDC is stopped.

To set the output timing, write data to TW0BFCM4 and TW0BFCM5.

$\overline{\text{RESET}}$ input or clearing the CE0 bit of TW0C sets these registers to 000H.

(5) 10-bit buffer registers 0 to 5 (TW0BFCM0 to TW0BFCM5)

TW0BFCM0 to TW0BFCM5 are 10-bit registers. They transfer data to the compare register (TW0CM0 to TW0CM5) corresponding to each buffer register at the timing of interrupt request signal INTTW0UD generation.

TW0BFCM0 to TW0BFCM5 can be read/written irrespective of whether TW0UDC count is stopped or operating.

$\overline{\text{RESET}}$ input sets TW0BFCM0 to TW0BFCM2, TW0BFCM4 and TW0BFCM5 to 000H, and TW0BFCM3 to 0FFH.

These registers can be read/written in word and byte units. For read/write operations of less than 8 bits, TW0BFCM0L to TW0BFCM5L are used.

(6) Dead-time reload register (TW0DTIME)

TW0DTIME is an 8-bit register to set dead time and is common to three dead-time timers (DTM0 to DTM2).

However, the data load timing from TW0DTIME to DTM0, DTM1 and DTM2 is independent.

TW0DTIME can be written only while TW0UDC counting is stopped. Data does not change even if an instruction to rewrite TW0DTIME is executed during timer operation.

$\overline{\text{RESET}}$ input sets TW0DTIME to FFH.

Even if TW0DTIME is set to 00H, an output with the dead time of 1/fx is performed.

(7) Dead-time timers 0 to 2 (DTM0 to DTM2)

DTM0 to DTM2 are 8-bit down counters that generate dead time.

Count down is performed after the value of the dead-time reload register (TW0DTIME) is reloaded with the timing of a compare match between TW0CM0 to TW0CM2 and TW0UDC. DTM0 to DTM2 generate an underflow signal when 00H changes to FFH and stop with FFH.

The count clock is fx.

DTM0 to DTM2 cannot be read/written.

$\overline{\text{RESET}}$ input or clearing the CE0 bit of TW0C sets these registers to FFH.

(8) Buffer transfer control timer (RTM0)

RTM0 is a 3-bit up counter. It has the function of dividing interrupt request signal INTTW0UD.

Incrementing is performed with the TW0UDC underflow signal and INTTW0UD is generated when the value matches the number of divisions set with bits IDEV00 to IDEV02 of TW0C.

RTM0 cannot be read/written.

$\overline{\text{RESET}}$ input sets RTM0 to 7H. Generating INTTW0UD and clearing the CE0 bit of TW0C also sets RTM0 to 7H.

6.4 Registers Controlling 10-Bit Inverter Control Timer

The following four registers control the 10-bit inverter control timer.

- Inverter timer control register (TW0C)
- Inverter timer mode register (TW0M)
- A/D trigger selection register (TW0TRGS)
- Inverter timer output control register (TW0OC)

(1) Inverter timer control register (TW0C)

TW0C controls the operation of TW0UDC, dead-time timers 0 to 2 (DTM0 to DTM2), and the buffer transfer control timer (RTM0), specifies the count clock of TW0UDC, and selects the compare register transfer cycle.

TW0C is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TW0C to 00H.

Figure 6-2. Format of Inverter Timer Control Register

Address: FF88H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TW0C	CE0	0	TCL02	TCL01	TCL00	IDEV02	IDEV01	IDEV00

CE0	TW0UDC, DTM0 to DTM2, RTM0 operation control
0	Clear and stop (TW0TO0 to TW0TO5 are Hi-Z)
1	Count enable

TCL02	TCL01	TCL00	Count clock selection	
				At fx = 20 MHz
0	0	0	fx	20 MHz
0	0	1	fx/2	10 MHz
0	1	0	fx/2^2	5 MHz
0	1	1	fx/2^3	2.5 MHz
1	0	0	fx/2^4	1.25 MHz
1	0	1	fx/2^5	625 kHz
Other than above			Setting prohibited	

IDEV02	IDEV01	IDEV00	INTTW0UD occurrence frequency selection
0	0	0	Occurs once every TW0UDC underflow.
0	0	1	Occurs once every two TW0UDC underflows.
0	1	0	Occurs once every three TW0UDC underflows.
0	1	1	Occurs once every four TW0UDC underflows.
1	0	0	Occurs once every five TW0UDC underflows.
1	0	1	Occurs once every six TW0UDC underflows.
1	1	0	Occurs once every seven TW0UDC underflows.
1	1	1	Occurs once every eight TW0UDC underflows.

Remark fx: System clock oscillation frequency

(2) Inverter timer mode register (TW0M)

TW0M controls the operation of and specifies the active level of the TW0TO0 to TW0TO5 outputs, and sets the valid edge of TW0TOFFP.

TW0M is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TW0M to 00H.

Figure 6-3. Format of Inverter Timer Mode Register

Address: FF89H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TW0M	0	0	0	PNOFFB0[Note]	ALV0	TOEDG0	TOSPP0	0

PNOFFB0[Note]	Control status flag output to TW0TO0 to TW0TO5
0	Output disabled status (TW0TO0 to TW0TO5 are Hi-Z)
1	Output enabled status

ALV0	TW0TO0 to TW0TO5 output active level specification
0	Low level
1	High level

TOEDG0	TW0TOFFP valid edge specification
0	Falling edge
1	Rising edge

TOSPP0	TW0TO0 to TW0TO5 output stop control by valid edge of TW0TOFFP
0	Output not stopped.
1	Output stopped (TW0TO0 to TW0TO5 are Hi-Z).

Note The PNOFFB0 bit is a read-only flag. This bit cannot be set or reset by software.

The PNOFFB0 bit is reset in following cases.

- When TW0UDC is stopped (CE0 = 0)
- When an output stop is generated by TW0TOFFP and INTWDT while TW0UDC is operating (CE0 = 1).

Caution Always set bits 0, 5 to 7 of TW0M to 0.

Remarks 1. TW0TO0 to TW0TO5 become Hi-Z state in the following cases. However, the TW0UDC, DTM0 to DTM2, and RTM0 timers do not stop if CE0 = 1 is set.

- A valid edge is input to the TW0TOFFP pin while TOSPP0 = 1.

To restore the output of TW0TO0 to TW0TO5, perform the procedure below.

<1> Write 0 to CE0 and stop the timer.
<2> Write 0 to the output stop function flag that is used.
<3> Reset the registers to their default values.

2. PNOFFB0, ALV0, CE0, and TW0TO0 to TW0TO5 are related as follows.

PNOFFB0	ALV0	CE0	TW0TO0, TW0TO2, TW0TO4	TW0TO1, TW0TO3, TW0TO5
0	0	0	Hi-Z	Hi-Z
0	1	0	Hi-Z	Hi-Z
0	0/1	1	Hi-Z	Hi-Z
1	0/1	1	PWM wave output	PWM wave output

(3) A/D trigger selection register (TW0TRGS)

TW0TRGS is a register used to select the A/D converter trigger signal from INTTW0CM4 and INTTW0CM5, which are generated upon a match between the compare register (TW0CM4, TW0CM5) and timer counter (TW0UDC).

TW0TRGS can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TW0TRGS to 00H.

Figure 6-4. Format of A/D Trigger Selection Register

Address: FF8BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TW0TRGS	0	0	0	0	0	0	TRSW1	TRSW0

TRSW1	TRSW0	Selection of A/D trigger
0	0	No output (INTADTR is kept "Low" level)
0	1	INTTW0CM4
1	0	INTTW0CM5
1	1	INTTW0CM4 or INTTW0CM5

(4) Inverter timer output control register (TW0OC)

TW0OC sets timer output stop in phase (U-phase/V-phase/W-phase) units.

TW0OC can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets TW0OC to 00H.

Figure 6-5. Format of Inverter Timer Output Control Register

Address: FF8CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TW0OC	0	0	0	0	0	0	TOSPW1	TOSPW0

TOSPW1	TOSPW0	Output control for PWM output
0	0	TW0TO0 to TW0TO5 output are permited
0	1	TW0TO0 and TW0TO1 output are prohibited (U phase off)
1	0	TW0TO2 and TW0TO3 output are prohibited (V phase off)
1	1	TW0TO4 and TW0TO5 output are prohibited (W phase off)

6.5 Registers Controlling 10-Bit Inverter Control Timer

(1) Setting procedure

(a) The TW0UDC count clock is set with the TCL00 to TCL02 bits of inverter timer control register (TW0C) and the occurrence frequency of interrupt request signal INTTW0UD is set with the IDEV00 to IDEV02 bits.

(b) The active level of the TW0TO0 to TW0TO5 pins is set with the ALV0 bit of inverter timer mode register (TW0M).

(c) Set the half width of the first PWM cycle to 10-bit compare register 3 (TW0CM3).

- PWM cycle = TW0CM3 value × 2 × TW0UDC clock rate
  (The clock rate of TW0UDC is set with the TW0C)

(d) Set the half width of the second PWM cycle to 10-bit buffer register 3 (TW0BFCM3).

(e) Set the dead time width to the dead time reload register (TW0DTIME).

- Dead time width = (TW0DTIME + 1) × 1/f_X
  f_X: Internal system clock

(f) Set the F/F set/reset timing that is used during the first cycle to 10-bit compare registers 0 to 2 (TW0CM0 to TW0CM2).

(g) Set the F/F set/reset timing that is used during the second cycle to TW0BFCM3.

(h) After the CE0 bit of TW0C is set (1), the operation of TW0UDC, dead-time timers 0 to 2 (DTM0 to DTM2), and buffer transfer control timer (RTM0) is enabled.

Caution Always use a bit manipulation instruction to set the CE0 bit.

(i) Set the F/F set/reset timing that is used for the next cycle to TW0BFCM0 to TW0BFCM5 during TW0UDC operation.

(j) To stop the TW0UDC operation, set the CE0 bit of the TW0C to 0.

Caution Another bit cannot be rewritten at the same time that the CE0 bit is being rewritten.

(2) Output waveform widths corresponding to set values

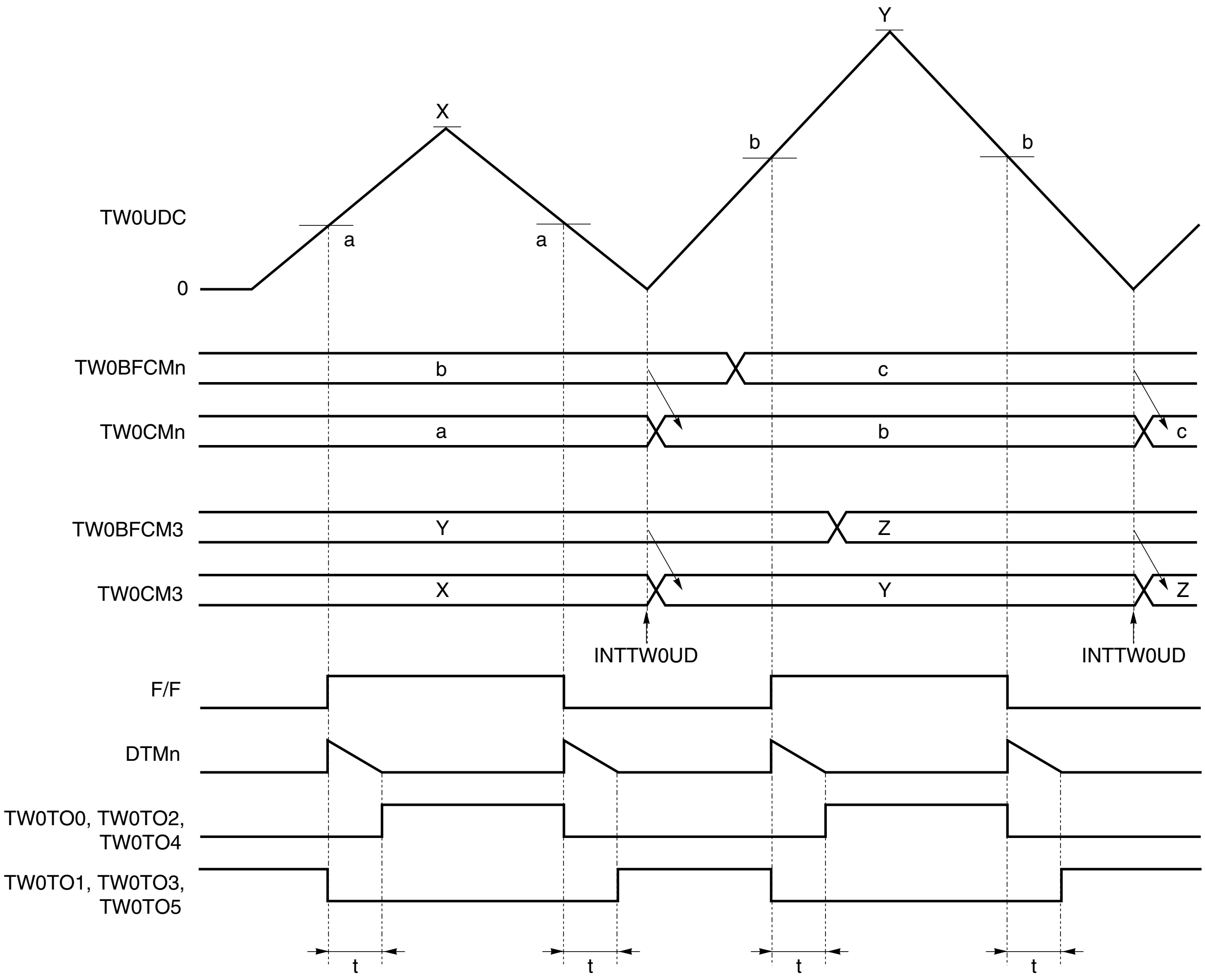
- PWM $_{\text{cycle}}$ = TW0CM3 × 2 × T_{TW0}
- Dead-time width = T_{DTM} = (TW0DTIME + 1) × 1/f_X
- Active width of positive phase (TW0TO0, TW0TO2, TW0TO4 pin)
  = {(TW0CM3 – TW0CM_{up}) + (TW0CM3 – $\text{TW0CM}_{\text{down}}$)} × T_{TW0} – T_{DTM}
- Active width of negative phase (TW0TO1, TW0TO3, TW0TO5 pin)
  = ($\text{TW0CM}_{\text{down}}$ + TW0CM_{up}) × T_{TW0} – T_{DTM}

f_X:	System clock oscillation frequency
T_{TW0}:	TW0UDC count clock
TW0CM_{up}:	Set value of TW0CM0 to TW0CM2 during TW0UDC count up
$\text{TW0CM}_{\text{down}}$:	Set value of TW0CM0 to TW0CM2 during TW0UDC count down

Caution If a value whose active width in the positive phase or negative phase becomes 0 or negative via the above calculation, TW0TO0 to TW0TO5 output a waveform fixed at the inactive level with an active width of 0 (refer to Figure 6-7).
However, if TW0CMn = 0 and TW0BFCMn ≥ TW0CM3 are set, TW0TO0 to TW0TO5 output a waveform at the active level.

(3) Operation timing

Figure 6-6. TW0UDC Operation Timing (Basic Operation)

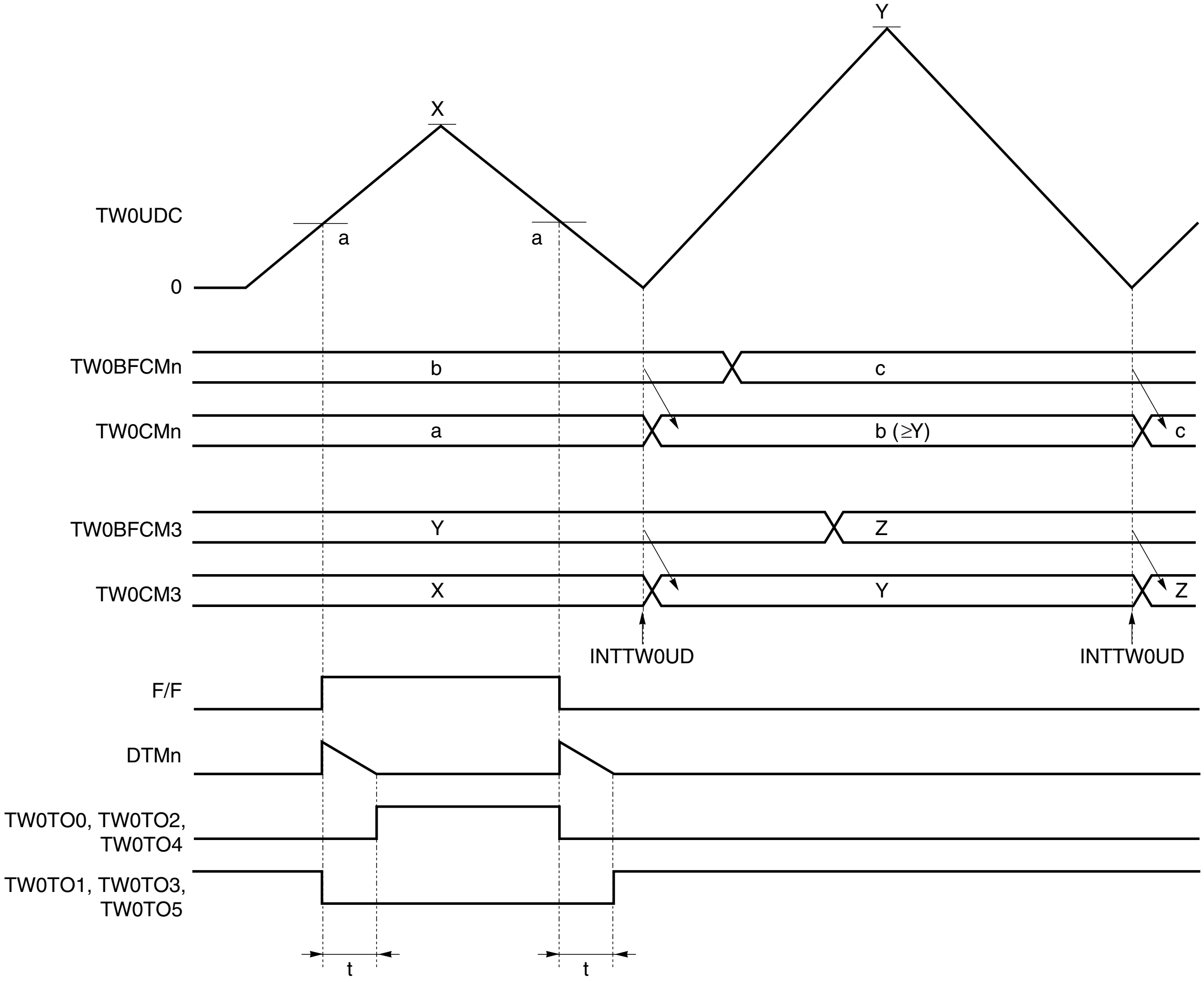


Remarks 1. n = 0 to 2

2. t: Dead time = (TW0DTIME + 1) × 1/fx
(fx: System clock oscillation frequency)

3. The above figure assumes an active high and undivided INTTW0UD occurrence.

Figure 6-7. TW0UDC Operation Timing (TW0CMn (TW0BFCMn) ≥ TW0CM3 (TW0BFCM3))



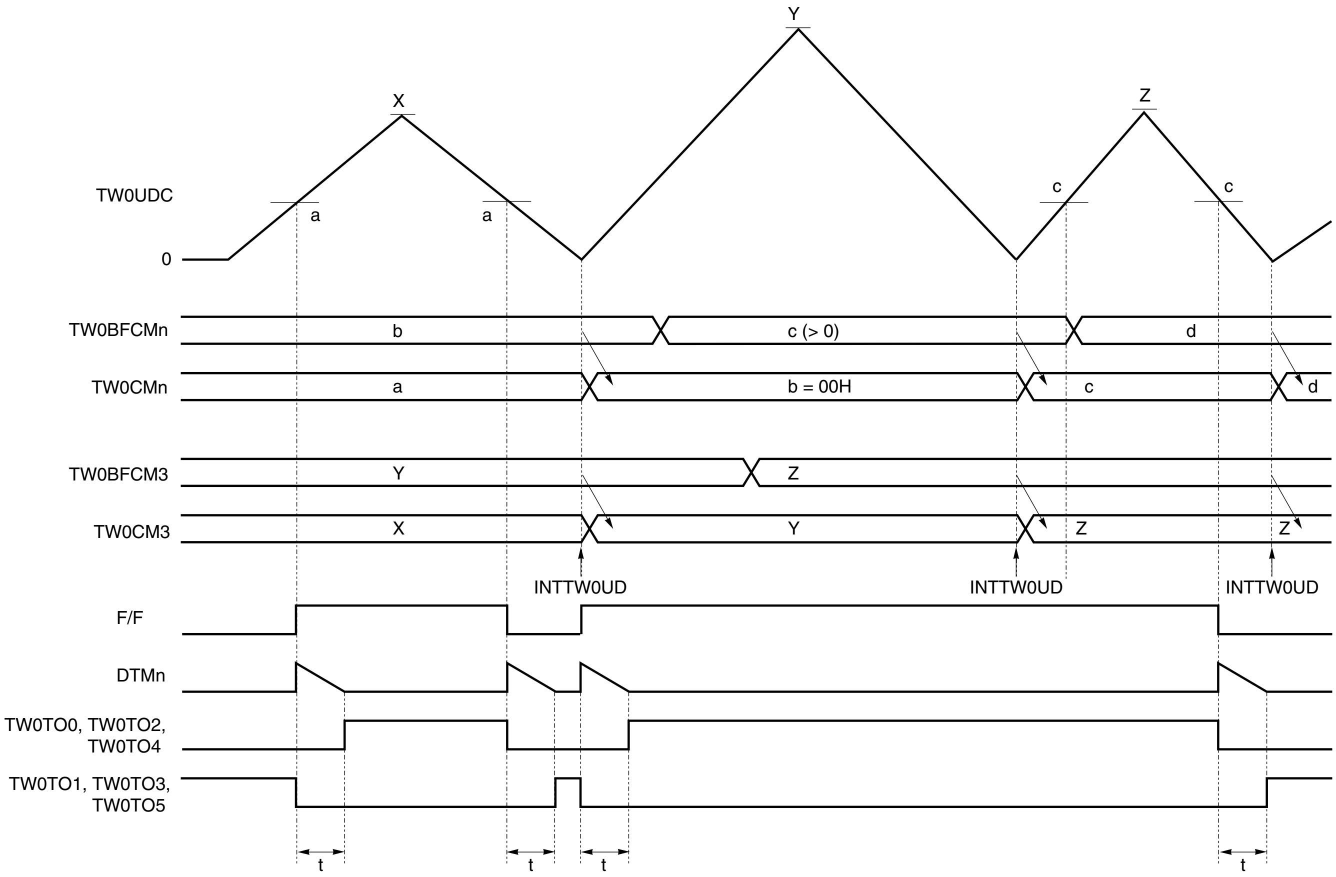
Remarks 1. n = 0 to 2

2. t: Dead time = (TW0DTIME + 1)) × 1/fx
(fx: System clock oscillation frequency)

3. The above figure assumes an active high and undivided INTTW0UD occurrence.

If a value higher than TW0CM3 is set to TW0BFCMn, low-level output in the positive phases (TW0TO0, TW0TO2, TW0TO4 pins), and high-level output in the negative phases (TW0TO1, TW0TO3, TW0TO5 pins) are continued. This setting is effective to output signals whose low and high widths are longer than the PWM cycle when controlling an inverter, etc.

Figure 6-8. TW0UDC Operation Timing (TW0CMn (TW0BFCMn) = 000H)

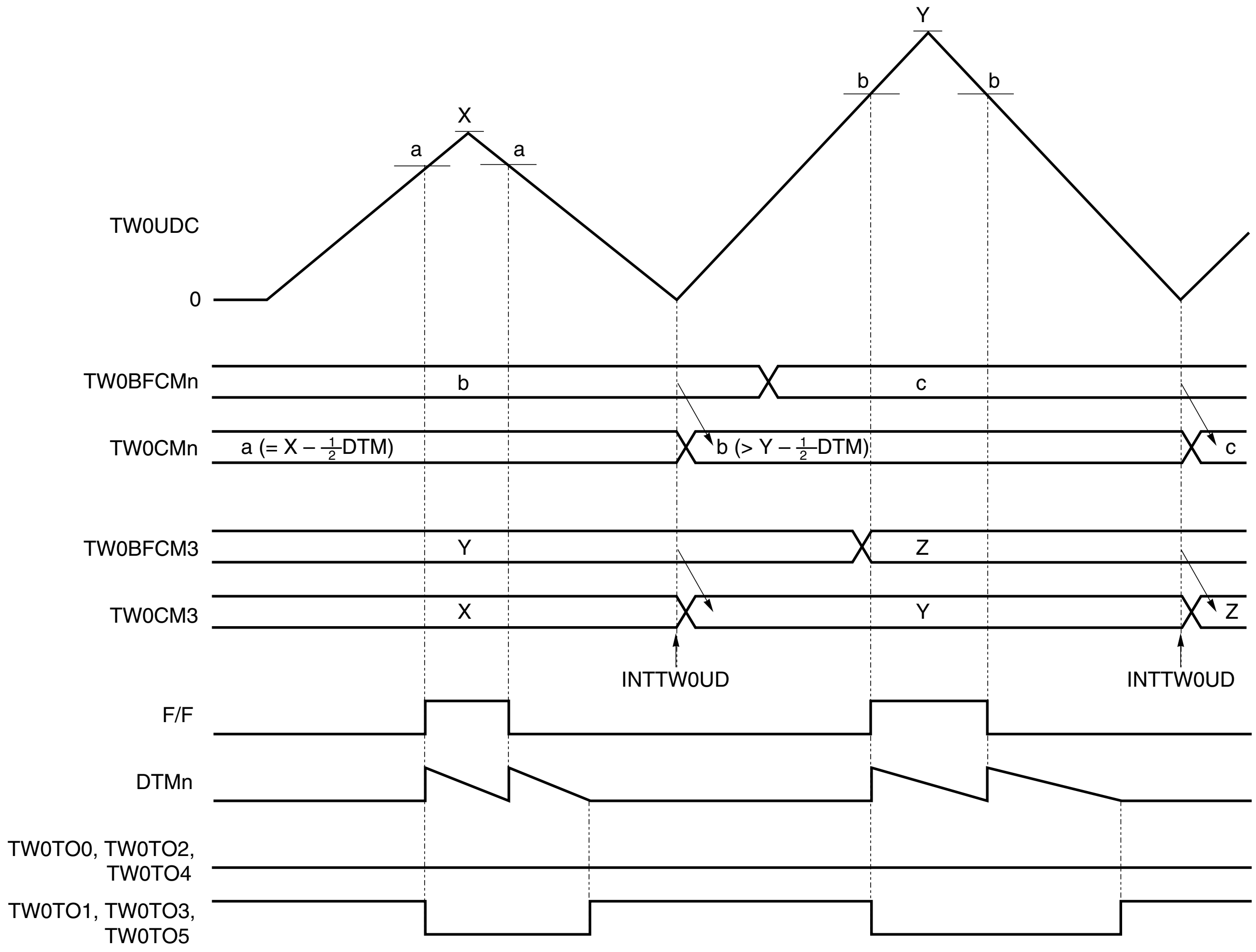


Remarks 1. n = 0 to 2

2. t: Dead time = (TW0DTIME + 1)) × 1/fx
(fx: System clock oscillation frequency)

3. The above figure assumes an active high and undivided INTTW0UD occurrence.

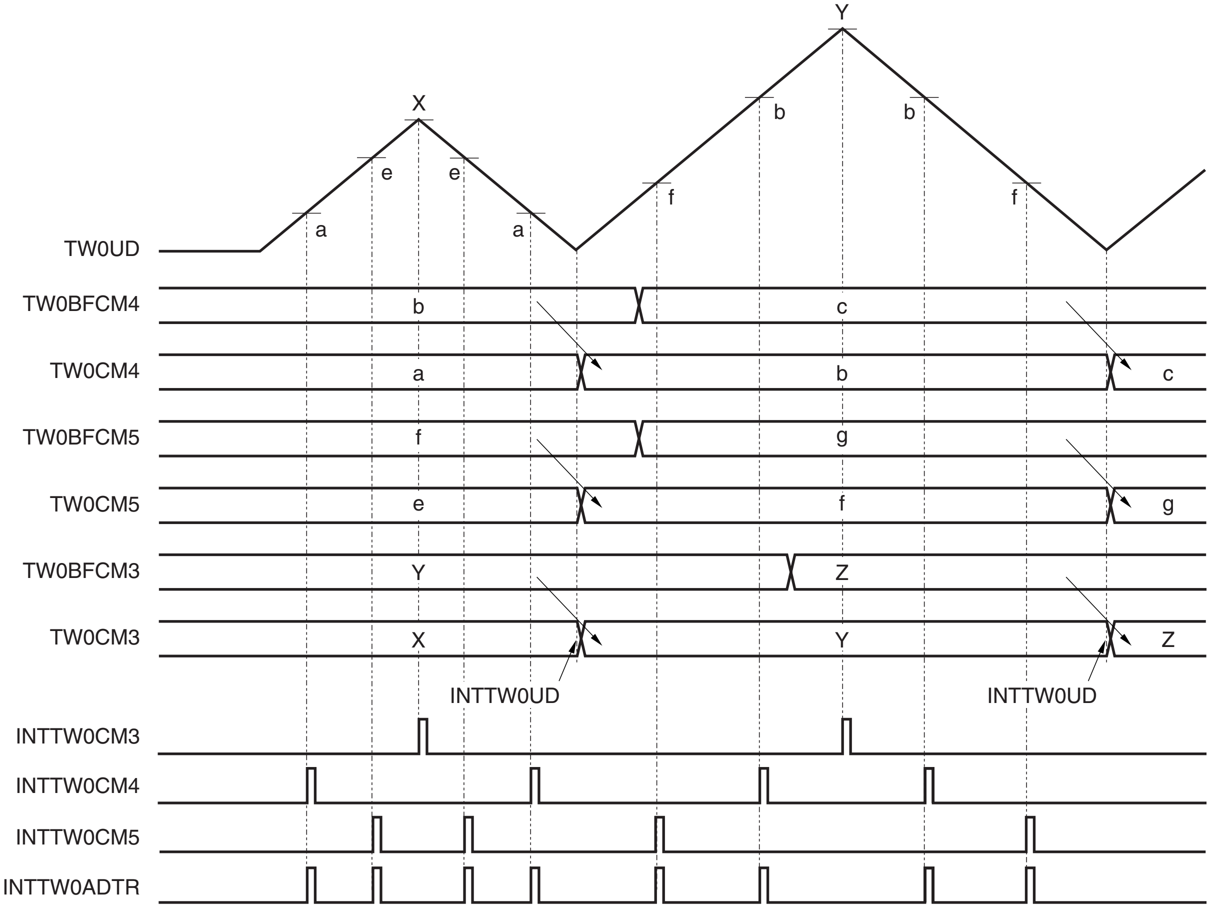
Figure 6-9. TW0UDC Operation Timing
(TW0CMn (TW0BFCMn) = TW0CM3 – 1/2DTM, TW0CMn (TW0BFCMn) > TW0CM3 – 1/2DTM)



Remarks 1. n = 0 to 2

2. The above figure assumes an active high and undivided INTTW0UD occurrence.

Figure 6-10. TW0UDC Operation Timing (IDEV02 to IDEV00 = 000B, TW0TRGS = 03H)



7.1 Functions of 16-Bit Up/Down Counter ITENC20

16-bit up/down counter ITENC20 has the following functions.

- General-purpose timer mode
  Free-running timer
  PWM output
- Up/down counter mode
  UDC mode A
  UDC mode B

- 16-bit 2-phase encoder input up/down counter & general-purpose timer (IT20UDC): 1 channel
- Compare registers: 2
- Capture/compare registers: 2
- Interrupt request sources
  - Capture/compare match interrupt: 2
  - Compare match interrupt request: 2
- Capture request signal: 2 types
  - The IT20UDC value can be latched using the valid edge of the TIT20CC0 and TIT20CC1 pins corresponding to the capture/compare register as the capture trigger.
- Count clock selectable through division by prescaler (set the frequency of the count clock to 10 MHz or less)
- Timer/count clock source: 2 types
  (external pulse input or internal system clock division)
- 2-phase encoder input
  The 2-phase external encoder signal is used as the count clock of the timer/counter via the external clock input pins (TIT20IUD, TIT20CUD). The counter mode can be selected from among the following four modes.

  - Mode 1: Counts the input pulses of the count pulse input pin (TIT20IUD).
    Up/down is specified by the level of the other input pin (TIT20CUD).
  - Mode 2: Counts up/down using the respective input pulses of the up count pulse input pin and down count pulse input pin.
  - Mode 3: Counts up/down using the phase relationship of the pulses input to the 2 pins.
  - Mode 4: Counts up/down using the phase relationship of the pulses input to the 2 pins. Counting is done using the respective rising and falling edges of the pulses.

- PWM output function
  In the general-purpose timer mode, 16-bit resolution PWM can be output from the TIT20TO pin.

- Timer clear

  The following timer clear operations are performed according to the mode that is used.

  (a) General-purpose timer mode: Timer clear operation is possible upon occurrence of match with IT20CM0 set value.

  (b) Up/down counter mode: The timer clear operation can be selected from among the following four conditions.

  (i) Timer clear performed upon occurrence of match with IT20CM0 set value during IT20UDC up count operation, and timer clear performed upon occurrence of match with IT20CM1 set value during IT20UDC down count operation.

  (ii) Timer clear performed only by external input.

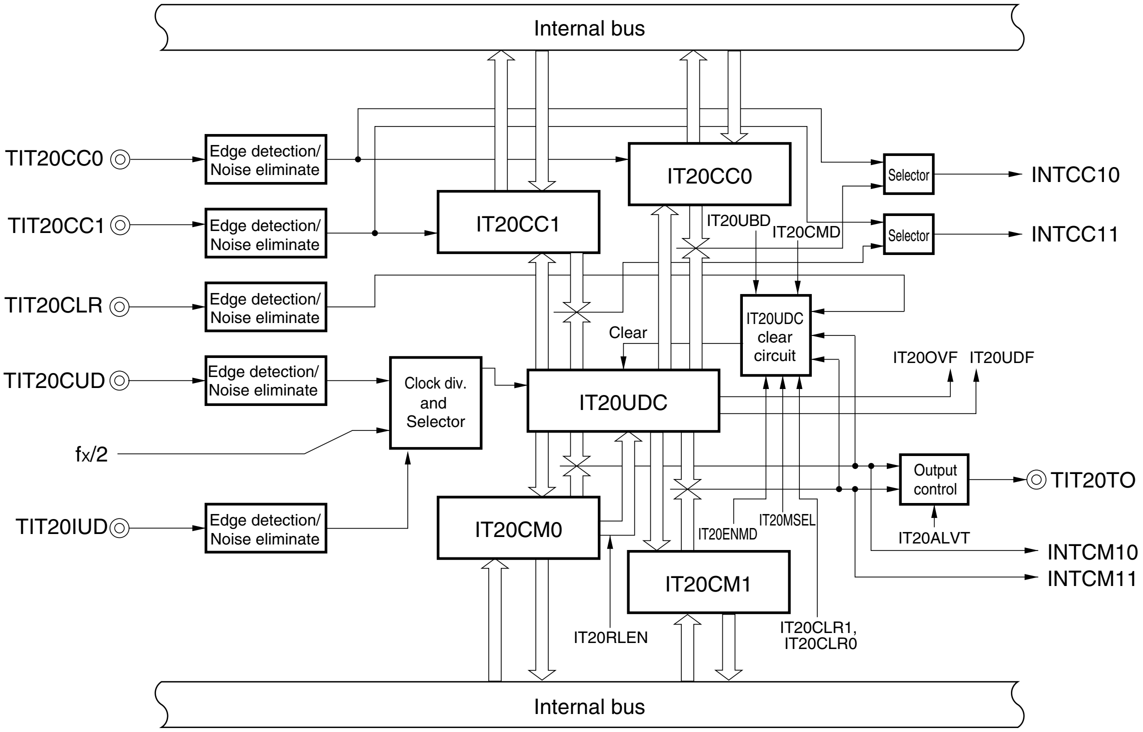
  (iii) Timer clear performed upon occurrence of match between IT20UDC count value and IT20CM0 set value.

  (iv) Timer clear performed upon occurrence of external input and match between IT20UDC count value and IT20CM0 set value.

- External pulse output (TIT20TO): 1

Figure 7-1 shows the block diagram of 16-bit up/down counter ITENC20.

Figure 7-1. Block Diagram of 16-bit Up/Down Counter ITENC20



7.2 Configuration of 16-bit Up/Down Counter ITENC20

16-bit Up/Down Counter ITENC20 consists of the following hardware.

Table 7-1. Configuration of 16-bit Up/Down Counter ITENC20

Item	Configuration	Generated Interrupt Signal	Capture Trigger
Timer counter	16-bit Up/Down Counter (IT20UDC)	–	–
Register	16-bit timer compare register 0 (IT20CM0)	INTCM10	–
	16-bit timer compare register 1 (IT20CM1)	INTCM11	–
	16-bit timer capture/compare register 0 (IT20CC0)	INTCC10	TIT20CC0
	16-bit timer capture/compare register 1 (IT20CC1)	INTCC11	TIT20CC1
Timer input	TIT20IUD, TIT20CUD, TIT20CC0, TIT20CC1, TIT20CLR	–	–
Timer output	TIT20TO	–	–
Control registers	Timer unit mode register (IT20TUM)	–	–
	Timer control register (IT20TMC)	–	–
	Capture/compare control register (IT20CCR)	–	–
	Valid edge select register (IT20SESA)	–	–
	Prescaler mode register (IT20PRM)	–	–
	Status register (IT20STS)	–	–
	Noise eliminate time select register 1 (NRC1)	–	–
	Port mode register 5 (PM5)	–	–
	Port register 5 (P5)	–	–

(1) 16-bit up/down counter (IT20UDC)

IT20UDC is a 2-phase encoder input up/down counter and general-purpose timer.

It can be read or written by a 16-bit memory manipulation instruction.

And, the lower 8 bits can be read or written with IT20UDCL by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20UDC to 0000H.

Address: FF11H, FF10H   After reset: 0000H   R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT20UDC																

Address: FF10H   After reset: 00H   R/W

Symbol	7	6	5	4	3	2	1	0
IT20UDCL								

Cautions 1. Writing to IT20UDC is enabled only when the IT20CE bit of the IT20TMC register is 0 (count operation disabled).

2. It is prohibited to set the IT20CMD bit (general-purpose timer mode) and the IT20MSEL bit (UDC mode B) of the IT20TUM register to 0 and 1, respectively.

3. Continuous reading of IT20UDC is prohibited. If IT20UDC is continuously read, the second read value may differ from the actual value. If IT20UDC must be read twice, be sure to read another register between the first and the second read operation.

4. Writing the same value to the IT20UDC, IT20CC0, and IT20CC1 registers, and the IT20STS register is prohibited.

Writing the same value to the IT20CCR, IT20TUM, IT20TMC, IT20SESA, and IT20PRM registers, and IT20CM0 and IT20CM1 registers is permitted (writing the same value is guaranteed even during a count operation).

IT20UDC start and stop is controlled by the IT20CE bit of timer control register (IT20TMC).

The IT20UDC operation consists of the following two modes.

(a) General-purpose timer mode

In the general-purpose timer mode, IT20UDC operates as a 16-bit interval timer, free-running timer, or PWM output.

Counting is performed based on the clock selected by software.

Division by the prescaler can be selected for the count clock from among fx/2, fx/4, fx/8, fx/16, fx/32, fx/64, or fx/128, using the IT20PRM2 to IT20PRM0 bits of prescaler mode register (IT20PRM) (fx: Internal system clock).

(b) Up/down counter mode (UDC mode)

In the UDC mode, IT20UDC functions as a 16-bit up/down counter that performs counting based on the TIT20CUD and TIT20IUD input signals.

Two operation modes can be set by the IT20MSEL bit of the IT20TUM register for this mode.

(i) UDC mode A (when IT20CMD bit = 1, IT20MSEL bit = 0)

IT20UDC can be cleared by setting the IT20CLR1 and IT20CLR0 bits of the IT20TMC register.

(ii) UDC mode B (when IT20CMD bit = 1, IT20MSEL bit = 1)

IT20UDC is cleared upon a match with IT20CM0 during an IT20UDC up count operation.

IT20UDC is cleared upon a match with IT20CM1 during an IT20UDC down count operation.

When the IT20CE bit of the IT20TMC register is 1, IT20UDC counts up when the operation mode is the general-purpose mode, and counts up/down when the operation mode is the UDC mode.

Cautions 1. TIT20CUD and TIT20CC0 are alternate-function pins. Therefore, when the TIT20CUD pin is used in the UDC mode, the external capture function of the TIT20CC0 pin cannot be used.

2. TIT20CLR and TIT20CC1 are alternate-function pins. Therefore, when the TIT20CLR input is used in UDC mode A, the external capture function of the TIT20CC1 pin cannot be used.

The conditions for clearing IT20UDC are as follows, according to the operation mode.

Table 7-2. Clear Conditions of 16-bit up/down counter (IT20UDC)

Operation Mode	IT20TUM Register		IT20TMC Register			IT20UDC Clear
	IT20CMD Bit	IT20MSEL Bit	IT20ENMD Bit	IT20CLR1 Bit	IT20CLR0 Bit	
General-purpose timer mode	0	0	0	×	×	Clearing not performed
			1	×	×	Cleared upon match with IT20CM0 set value
UDC mode A	1	0	×	0	0	Cleared only by TIT20CLR input
			×	0	1	Cleared upon match with IT20CM0 set value during up count operation
			×	1	0	Cleared by TIT20CLR input or upon match with IT20CM0 set value during up count operation
			×	1	1	Clearing not performed
UDC mode B	1	1	×	×	×	Cleared upon match with IT20CM0 set value during up count operation or upon match with IT20CM1 set value during down count operation
Other than above						Setting prohibited

Remark ×: Indicates that the set value of that bit is ignored.

(2) Compare register 0 (IT20CM0)

IT20CM0 is a 16-bit register that always compares its value with the value of IT20UDC. When the value of the compare register matches the value of IT20UDC, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In the general-purpose timer mode (IT20CMD bit of IT20TUM register = 0) and UDC mode A (IT20MSEL bit of IT20TUM register = 0), an interrupt signal (INTCM10) is always generated upon occurrence of a match.
- In UDC mode B (IT20MSEL bit of IT20TUM register = 1), an interrupt signal (INTCM10) is generated only upon occurrence of a match during a down count operation.

IT20CM0 can be read or written by a 16-bit memory manipulation instruction.
And, the lower 8 bits can be read or written with IT20CM0L by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IT20CM0 to 0000H.

Address: FF13H, FF12H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT20CM0																

Address: FF12H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IT20CM0L								

Caution When the IT20CE bit of the IT20TMC register is 1, it is prohibited to overwrite the value of the IT20CM0 register.

(3) Compare register 1 (IT20CM1)

IT20CM1 is a 16-bit register that always compares its value with the value of IT20UDC. When the value of the compare register matches the value of IT20UDC, an interrupt signal is generated. The interrupt generation timing in the various modes is described below.

- In the general-purpose timer mode (IT20CMD bit of IT20TUM register = 0) and UDC mode A (IT20MSEL bit of IT20TUM register = 0), an interrupt signal (INTCM11) is always generated upon occurrence of a match.
- In UDC mode B (IT20MSEL bit of IT20TUM register = 1), an interrupt signal (INTCM11) is generated only upon occurrence of a match during a down count operation.

IT20CM1 can be read or written by a 16-bit memory manipulation instruction.
And, the lower 8 bits can be read or written with IT20CM1L by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IT20CM1 to 0000H.

Address: FF15H, FF14H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT20CM1																

Address: FF14H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IT20CM1L								

Caution When the IT20CE bit of the IT20TMC register is 1, it is prohibited to overwrite the value of the IT20CM1 register.

(4) Capture/compare register 0 (IT20CC0)

IT20CC0 is a 16-bit register. It can be specified as a capture register or as a compare register using capture/compare control register (IT20CCR).

IT20CC0 can be read or written by a 16-bit memory manipulation instruction.

And, the lower 8 bits can be read or written with IT20CC0L by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20CC0 to 0000H.

Address: FF91H, FF90H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT20CC0																

Address: FF90H After reset: 0000H R/W

Symbol	7	6	5	4	3	2	1	0
IT20CC0L								

Cautions 1. When used as a capture register (IT20CMS0 bit of IT20CCR register = 0), write access from the CPU is prohibited.

2. When used as a compare register (IT20CMS0 bit of IT20CCR register = 1) and the IT20CE bit of the IT20TMC register is 1, overwriting the IT20CC0 register values is prohibited.

3. When the IT20CE bit of the IT20TMC register is 0, the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, set a new compare value.

5. Continuous reading of IT20CC0 is prohibited. If IT20CC0 is continuously read, the second read value may differ from the actual value. If IT20CC0 must be read twice, be sure to read another register between the first and the second read operation.

(a) When set as a capture register

When IT20CC0 is set as a capture register, the valid edge of the corresponding TIT20CC0 signal is detected as the capture trigger. IT20UDC latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupts (rising edge, falling edge, both edges) is selected by valid edge select register (IT20SESA).

When the IT20CC0 register is specified as a capture register, interrupts are generated upon detection of the valid edge of the TIT20CC0 signal.

Caution TIT20CUD and TIT20CC0 are alternate-function pins. Therefore, when the TIT20CUD pin is used in the UDC mode, the external capture function of the TIT20CC0 pin cannot be used.

(b) When set as a compare register

When IT20CC0 is set as a compare register, it always compares its own value with the value of IT20UDC. If the value of IT20CC0 matches the value of the IT20UDC, IT20CC0 generates an interrupt signal (INTCC10).

(5) Capture/compare register 1 (IT20CC1)

IT20CC1 is a 16-bit register. It can be specified as a capture register or as a compare register using capture/compare control register (IT20CCR).

IT20CC1 can be read or written by a 16-bit memory manipulation instruction.

And, the lower 8 bits can be read or written with IT20CC1L by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20CC1 to 0000H.

Address: FF93H, FF92H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IT20CC1																

Address: FF92H After reset: 0000H R/W

Symbol	7	6	5	4	3	2	1	0
IT20CC1L								

Cautions 1. When used as a capture register (IT20CMS1 bit of IT20CCR register = 0), write access from the CPU is prohibited.

2. When used as a compare register (IT20CMS1 bit of IT20CCR register = 1) and the IT20CE bit of the IT20TMC register is 1, overwriting the IT20CC1 register values is prohibited.

3. When the IT20CE bit of the IT20TMC register is 0, the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, newly set a compare value.

5. Continuous reading of IT20CC1 is prohibited. If IT20CC1 is continuously read, the second read value may differ from the actual value. If IT20CC1 must be read twice, be sure to read another register between the first and the second read operation.

(a) When set as a capture register

When IT20CC1 is set as a capture register, the valid edge of the corresponding TIT20CC1 signal is detected as the capture trigger. IT20UDC latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupts (rising edge, falling edge, both edges) is selected by valid edge select register (IT20SESA).

When the IT20CC1 register is specified as a capture register, interrupts are generated upon detection of the valid edge of the TIT20CC1 signal.

Caution TIT20CLR and TIT20CC1 are alternate-function pins. Therefore, when the TIT20CLR input is used in UDC mode A, the external capture function of the TIT20CC1 pin cannot be used.

(b) When set as a compare register

When IT20CC1 is set as a compare register, it always compares its own value with the value of IT20UDC. If the value of IT20CC1 matches the value of the IT20UDC, IT20CC1 generates an interrupt signal (INTCC11).

7.3 16-Bit Up/down Counter ITENC20 Control Registers

The 16-Bit Up/down Counter ITENC20 is controlled by the following nine registers.

- Timer unit mode register (IT20TUM)
- Timer control register (IT20TMC)
- Capture/compare control register (IT20CCR)
- Valid edge select register (IT20SESA)
- Prescaler mode register (IT20PRM)
- Status register (IT20STS)
- Noise eliminate time select register 1 (NRC1)
- Port mode register 5 (PM5)
- Port register 5 (P5)

(1) Timer unit mode register (IT20TUM)

The IT20TUM register is an 8-bit register used to specify the IT20UDC operation mode or to control the operation of the PWM output pin.

This register can be read or written by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20TUM to 00H.

Figure 7-2. Format of Timer Unit Mode Register (IT20TUM)

Address: FF95H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
IT20TUM	IT20CMD	0	0	0	IT20TOE	IT20ALVT	0	IT20MSEL

IT20CMD	IT20UDC operation mode specification
0	General-purpose timer mode (up count)
1	UDC mode (up/down count)

IT20TOE	Specification of timer output (TIT20TO) enable
0	Timer output disabled
1	Timer output enabled
When IT20CMD bit = 1 (UDC mode), timer output is not performed regardless of the setting of the IT20TOE bit. At this time, timer output is the inverted phase level of the level set by the IT20ALVT bit.	

IT20ALVT	Specification of timer output (TIT20TO) active level
0	Active level is high level
1	Active level is low level
When IT20CMD bit = 1 (UDC mode), timer output is not performed regardless of the setting of the IT20TOE bit. At this time, timer output is the inverted phase level of the level set by the IT20ALVT bit.	

IT20MSEL	Specification of operation in UDC mode (up/down count).
0	UDC mode A IT20UDC can be cleared by setting the IT20CLR1 and IT20CLR0 bits of the IT20TMC register.
1	UDC mode B IT20UDC is cleared in the following cases. • Upon match with IT20CM0 during IT20UDC up count operation • Upon match with IT20CM1 during IT20UDC down count operation
When UDC mode B is set, the IT20ENMD, IT20CLR1, and IT20CLR0 bits of the IT20TMC register become invalid.	

Cautions 1. Changing the value of the IT20TUM register during IT20UDC operation (IT20CE bit of IT20TMC register = 1) is prohibited.

2. When the IT20CMD bit = 0 (general-purpose timer mode), setting IT20MSEL = 1 (UDC mode B) is prohibited.

(2) Timer control register (IT20TMC)

The IT20TMC register is used to enable/disable IT20UDC operation and to set transfer and timer clear operations.

This register can be read or written by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20TMC to 00H.

Figure 7-3. Format of Timer Control Register (IT20TMC)

Address: FF96H After reset: 00H R/W

Symbol	7	<6>	5	4	3	2	1	0
IT20TMC	0	IT20CE	0	0	IT20RLEN	IT20ENMD	IT20CLR1	IT20CLR0

IT20CE	IT20UDC operation control
0	Count operation disabled
1	Count operation enabled

<table>
<tr><th>IT20RLEN</th><th>Specification of transfer operation from IT20CM0 to IT20UDC</th></tr>
<tr><td>0</td><td>Transfer operation disabled</td></tr>
<tr><td>1</td><td>Transfer operation enabled</td></tr>
<tr><td colspan="2">• When IT20RLEN = 1, the value set to IT20CM0 is transferred to IT20UDC upon occurrence of a IT20UDC underflow.
• When the IT20CMD bit of the IT20TUM register = 0 (general-purpose timer mode), the IT20RLEN bit setting becomes invalid.
• The IT20RLEN bit is valid only in UDC mode A (IT20TUM register’s IT20CMD bit = 1, IT20MSEL bit = 0). In the general-purpose timer mode (IT20CMD bit = 0) and in UDC mode B (IT20CMD bit = 1, IT20MSEL bit =1), a transfer operation is not performed even if the IT20RLEN bit is set (1).</td></tr>
</table>

<table>
<tr><th>IT20ENMD</th><th>Control of IT20UDC clear operation in general-purpose timer mode</th></tr>
<tr><td>0</td><td>Clear disabled (free-running mode)
Clearing is not performed even when IT20UDC and IT20CM0 values match</td></tr>
<tr><td>1</td><td>Clear enabled
Clearing is performed when IT20UDC and IT20CM0 values match.</td></tr>
<tr><td colspan="2">When the IT20CMD bit of the IT20TUM register = 1 (UDC mode), the IT20ENMD bit setting becomes invalid.</td></tr>
</table>

<table>
<tr><th>IT20CLR1</th><th>IT20CLR0</th><th>IT20UDC clear source specification</th></tr>
<tr><td>0</td><td>0</td><td>Cleared only by external input (TIT20CLR)</td></tr>
<tr><td>0</td><td>1</td><td>Cleared upon match of IT20UDC count value and IT20CM0 set value</td></tr>
<tr><td>1</td><td>0</td><td>Cleared by TIT20CLR input or upon match of IT20UDC count value and IT20CM0 set value</td></tr>
<tr><td>1</td><td>1</td><td>Not cleared</td></tr>
<tr><td colspan="3">• Clearing by match of the IT20UDC count value and IT20CM0 set value is valid only during a IT20UDC up count operation (IT20UDC is not cleared during a IT20UDC down count operation).
• When the IT20CMD bit of the IT20TUM register = 0 (general-purpose timer mode), the IT20CLR1 and IT20CLR0 bit settings are invalid.
• When the MSEL0 bit of the IT20TUM register = 1 (UDC mode B), the IT20CLR1 and IT20CLR0 bit settings are invalid.
• When clearing by TIT20CLR has been enabled by bits IT20CLR1 and IT20CLR0, clearing is performed regardless of whether the value of the IT20CE bit is 1 or 0.</td></tr>
</table>

Caution Changing the values of the IT20TMC register bits other than the IT20CE bit during IT20UDC operation (IT20CE = 1) is prohibited.

(3) Capture/compare control register (IT20CCR)

The IT20CCR register specifies the operation mode of the capture/compare registers (IT20CC0, IT20CC1).

This register can be read or written by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20CCR to 00H.

Figure 7-4. Format of Capture/Compare Control Register (IT20CCR)

Address: FF94H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
IT20CCR	0	0	0	0	0	0	IT20CMS1	IT20CMS0

IT20CMS1	IT20CC1 operation mode specification
0	Operates as capture register
1	Operates as compare register

IT20CMS0	IT20CC0 operation mode specification
0	Operates as capture register
1	Operates as compare register

Cautions 1. Overwriting the IT20CCR register during IT20UDC operation (IT20CE bit = 1) is prohibited.

2. TIT20CUD and TIT20CC0 are alternate-function pins. Therefore, when the TIT20CUD pin is used in the UDC mode, the external capture function of the TIT20CC0 pin cannot be used.

3. TIT20CLR and TIT20CC1 are alternate-function pins. Therefore, when the TIT20CLR input is used in UDC mode A, the external capture function of the TIT20CC1 pin cannot be used.

(4) Valid edge select register (IT20SESA)

The IT20SESA register is used to specify the valid edge of external interrupt requests from the external pins (TIT20CC0, TIT20CC1, TIT20IUD, TIT20CUD, TIT20CLR).

The valid edge (rising edge, falling edge, or both edges) can be specified independently for each pin.

This register can be read or written by a 1-bit or 8-bit memory manipulation instruction.

RESET input sets IT20SESA to 00H.

Figure 7-5. Format of Valid Edge Select Register (IT20SESA)

Address: FF97H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IT20SESA	IT20TESUD1	IT20TESUD0	IT20CESUD1	IT20CESUD0	IT20IES11	IT20IES10	IT20IES01	IT20IES00
	TIT20IUD, TIT20CUD		TIT20CLR		TIT20CC1		TIT20CC0	

IT20TESUD1	IT20TESUD0	Specification of valid edge of TIT20IUD and TIT20CUD pins
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges
• The set values of the IT20TESUD1 and IT20TESUD0 bits are only valid in UDC mode A and UDC mode B. • If mode 4 is specified as the operation mode of IT20UDC (specified by the IT20PRM2 to IT20PRM0 bits of the IT20PRM register), the valid edge specifications for the TIT20IUD and TIT20CUD pins (IT20TESUD1 and IT20TESUD0 bits) are not valid.		

IT20CESUD1	IT20CESUD0	Specification of valid edge of TIT20CLR pin
0	0	Falling edge (IT20UDC cleared after edge detection)
0	1	Rising edge (IT20UDC cleared after edge detection)
1	0	Low level (IT20UDC cleared status held)
1	1	High level (IT20UDC cleared status held)
• The set values of the IT20CESUD1 andIT20CESUD0 bits are valid only in UDC mode A.		

IT20IES11	IT20IES10	Specification of valid edge of TIT20CC1 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

IT20IES01	IT20IES00	Specification of valid edge of TIT20CC0 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

Caution Changing the values of the IT20SESA register bits during IT20UDC operation (IT20CE = 1) is prohibited.

(5) Prescaler mode register (IT20PRM)

The IT20PRM register is used to perform the following selections.

- Selection of count clock in general-purpose timer mode (IT20CMD bit of IT20TUM register = 0)
- Selection of count operation mode in UDC mode (IT20CMD = 1)

This register can be read or written by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20PRM to 07H.

Figure 7-6. Format of Priscaler Mode Register (IT20PRM)

Address: FF3AH After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
IT20PRM	0	0	0	0	0	IT20PRM2	IT20PRM1	IT20PRM0

IT20PRM2	IT20PRM1	IT20PRM0	IT20CMD = 0	IT20CMD = 1	
			Count clock	Count clock	Up/down count
0	0	0	Setting prohibited	Setting prohibited (Mode 4) (At this time, the IT20SESA register is enabled.)	
0	0	1	fx/2		
0	1	0	fx/4		
0	1	1	fx/8		
1	0	0	fx/16	TIT20IUD	Mode 1
1	0	1	fx/32		Mode 2
1	1	0	fx/64		Mode 3
1	1	1	fx/128		Mode 4

Remark fx: Internal system clock

Cautions 1. Overwriting the IT20PRM register during IT20UDC operation (IT20CE bit = 1) is prohibited.

2. When the IT20CMD bit of the IT20TUM register = 1 (UDC mode), setting the values of the IT20PRM2 to IT20PRM0 to 000, 001, 010, and 011 bits is prohibited.

3. When IT20UDC is in mode 4, specification of the valid edge for the TIT20IUD and TIT20CUD pins is invalid.

(a) In general-purpose timer mode (IT20CMD bit of IT20TUM register = 0)

The count clock is fixed to the internal clock. The clock rate of IT20UDC is specified by the IT20PRM2 to IT20PRM0 bits.

(b) UDC mode (IT20CMD bit of IT20TUM register = 1)

The IT20UDC count triggers in the UDC mode are as follows.

Operation Mode	IT20UDC Operation
Mode 1	Down count when TIT20CUD = high level Up count when TIT20CUD = low level
Mode 2	Up count upon detection of valid edge of TIT20IUD input Down count upon detection of valid edge of TIT20CUD input
Mode 3	Automatic judgment by TIT20CUD input level upon detection of valid edge of TIT20IUD input
Mode 4	Automatic judgment upon detection of both edges of TIT20IUD input and both edges of TIT20CUD input

(6) Status register (IT20STS)

The IT20STS register indicates the operating status of IT20UDC.

This register can be read by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets IT20STS to 00H.

Figure 7-7. Format of Status Register (IT20STS)

Address: FF3BH After reset: 00H R

Symbol	7	6	5	4	3	<2>	<1>	<0>
IT20STS	0	0	0	0	0	IT20UDF	IT20OVF	IT20UBD

IT20UDF	IT20UDC underflow flag
0	No IT20UDC count underflow
1	IT20UDC count underflow
The IT20UDF bit is cleared (0) upon completion of a read access to the IT20STS register from the CPU.	

IT20OVF	IT20UDC overflow flag
0	No IT20UDC count overflow
1	IT20UDC count overflow
The IT20OVF bit is cleared (0) upon completion of a read access to the IT20STS register from the CPU.	

IT20UBD	IT20UDC up/down count operation status
0	IT20UDC up count in progress
1	IT20UDC down count in progress
The state of the IT20UBD bit differs according to the mode as follows. • The IT20UBD bit is fixed to 0 by hardware when the IT20CMD bit of the IT20TUM register = 0 (general-purpose timer mode). • The IT20UBD bit indicates the IT20UDC up/down count status when the IT20CMD bit of the IT20TUM register = 1 (UDC mode).	

Caution Overwriting the IT20STS register during IT20UDC operation (IT20CE bit = 1) is prohibited.

(7) Noise eliminate time select register 1 (NRC1)

The NRC1 register selects the sampling clock that is used to eliminate digital noise on the TIT20IUD, TIT20CUD, TIT20CC0, TIT20CC1, or TIT20CLR pin. If a level is not detected on these pins five times in a row at the clock selected by the NRC1 register, the signal is eliminated as noise.

This register can be read or written by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 7-8. Format of Noise Eliminate Time Select Register 1 (NRC1)

Address: FFAAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NRC1	0	0	0	0	0	0	NRC11	NRC10

NRC11	NRC10	Sampling clock selection
0	0	fx/2^3
0	1	fx/2^2
1	0	fx/2
1	1	fx

Remark fx: Internal system clock

Cautions 1. If the input pulse lasts for the duration of 4 to 5 clocks, it is undefined whether the pulse is detected as a valid edge or eliminated as noise. So that the pulse is actually detected as a valid edge, a pulse level must be input for the duration of 5 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

(8) Port mode register 5 (PM5)

This register sets port 5 input/output in 1-bit units.

When using the P57/TIT20CLR/TIT20CC1/TIT20TO pin for timer output, clear PM57 and the output latch of P57 to 0.

When using the P55/TIT20IUD/INTP6, P56/TIT20CUD/TIT20CC0/INTP7, and P57/TIT20CLR/TIT20CC1/TIT20TO pins for timer input, set PM55, PM56, and PM57 to 1. The output latches of P55, P56, and P57 at this time may be 0 or 1.

PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to FFH.

Figure 7-9. Format of Port Mode Register 5 (PM5)

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	PM5n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 16-Bit Up/down Counter ITENC20 Operations

7.4.1 Basic operation

The following two operation modes can be selected for 16-bit up/down counter ITENC20.

(1) General-purpose timer mode (IT20CMD bit of IT20TUM register = 0)

In the general-purpose timer mode, this counter operates either as a 16-bit interval timer or as a PWM output timer (the count operation is up count only).

The count clock to IT20UDC is selected by prescaler mode register (IT20PRM).

(2) Up/down counter mode (UDC mode) (IT20CMD bit of IT20TUM register = 1)

In the UDC mode, this counter operates as a 16-bit up/down counter.

The external clock input (TIT20IUD, TIT20CUD pins) by IT20PRM register setting is used as the IT20UDC count clock.

The UDC mode is further divided into two modes according to the IT20UDC clear conditions.

- **UDC mode A (IT20TUM register's IT20CMD bit = 1, IT20MSEL bit = 0)**

  The IT20UDC clear source can be selected as only external clear input (TIT20CLR), a match signal between the IT20UDC count value and the IT20CM0 set value during up count operation, or the logical sum (OR) of the two signals, using the IT20CLR1 and IT20CLR0 bits of the IT20TMC register. IT20UDC can reload the value of IT20CM0 upon occurrence of an IT20UDC underflow.

- **UDC mode B (IT20TUM register's IT20CMD bit = 1, IT20MSEL bit = 1)**

  The status of IT20UDC after a match of the IT20UDC count value and IT20CM0 set value is as follows.

  <1> In the case of an up count operation, IT20UDC is cleared (0000H), and the INTCM10 interrupt is generated.

  <2> In the case of a down count operation, the IT20UDC count value is decremented (–1).

  The status of IT20UDC after a match of the IT20UDC count value and IT20CM1 set value is as follows.

  <1> In the case of an up count operation, the IT20UDC count value is incremented (+1).

  <2> In the case of a down count operation, IT20UDC is cleared (0000H), and the INTCM11 interrupt is generated.

7.4.2 Operation in general-purpose timer mode

16-bit up/down counter ITENC20 can perform the following operations in the general-purpose timer mode.

(1) Interval operation

IT20UDC and IT20CM0 always compare their values and the INTCM10 interrupt is generated upon occurrence of a match.

IT20UDC is cleared (0000H) at the count clock following the match.

Furthermore, when one more count clock is input, IT20UDC counts up to 0001H. The interval time can be calculated with the following formula.

$$\boxed{\text{Interval time} = (\text{IT20CM0 value} + 1) \times \text{IT20UDC count clock rate}}$$

Caution Interval operation can be achieved by setting the IT20ENMD bit of the IT20TMC register to 1.

(2) Free-running operation

IT20UDC performs a full count operation from 0000H to FFFFH, and after the IT20OVF bit of the IT20STS register is set (to 1), IT20UDC is cleared and resumes counting. The free-running cycle can be calculated by the following formula.

$$\boxed{\text{Free-running cycle} = 65{,}536 \times \text{IT20UDC count clock rate}}$$

Caution The free-running operation can be achieved by setting the IT20ENMD bit of the IT20TMC register to 0.

(3) Compare function

IT20UDC connects two compare register (IT20CM0, IT20CM1) channels and two capture/compare register (IT20CC0, IT20CC1) channels.

When the IT20UDC count value and the set value of one of the compare registers match, a match interrupt (INTCM10, INTCM11, INTCC10$^{\text{Note}}$, INTCC11$^{\text{Note}}$) is output.

Particularly in the case of interval operation, IT20UDC is cleared upon generation of the INTCM10 interrupt.

Note This match interrupt is generated when IT20CC0 and IT20CC1 are set to the compare register mode.

(4) Capture function

IT20UDC connects two capture/compare register (IT20CC0, IT20CC1) channels.

When IT20CC0 and IT20CC1 are set to the capture register mode, the value of IT20UDC is captured in synchronization with the corresponding capture trigger signal.

Furthermore, an interrupt request (INTCC10, INTCC11) is generated by the TIT20CC0 and TIT20CC1 input signals.

Table 7-3. Capture Trigger Signal to 16-Bit Capture Register

Capture Register	Capture Trigger Signal
IT20CC0	TIT20CC0
IT20CC1	TIT20CC1

Remark IT20CC0 and IT20CC1 are capture/compare registers. Which of these registers is used is specified by capture/compare control register (IT20CCR).

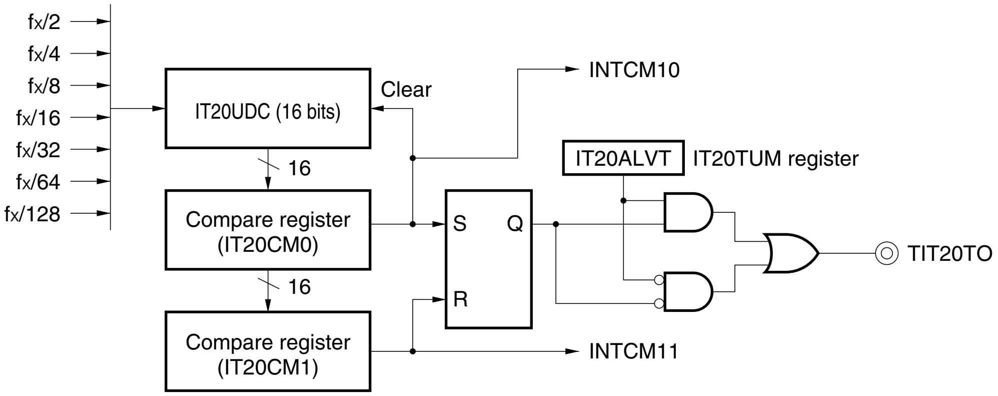
The valid edge of the capture trigger is specified by valid edge select register (IT20SESA). If both the rising edge and the falling edge are selected as the capture triggers, it is possible to measure the input pulse width externally. If a single edge is selected as the capture trigger, the input pulse cycle can be measured.

(5) PWM output operation

PWM output operation is performed from the TIT20TO pin by setting IT20UDC to the general-purpose timer mode (IT20CMD bit = 0) using timer unit mode register (IT20TUM).

The resolution is 16 bits, and the count clock can be selected from among seven internal clocks (fx/2, fx/4, fx/8, fx/16, fx/32, fx/64, fx/128).

Figure 7-10. Block Diagram During PWM Output Operation



Caution Be sure to set the count clock of IT20UDC to 10 MHz or lower.

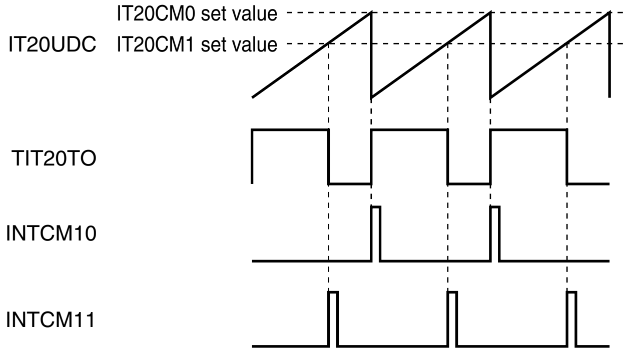
Remark fx: Internal system clock

(a) Description of operation

The IT20CM0 register is a compare register used to set the PWM output cycle. When the value of this register matches the value of IT20UDC, the INTCM10 interrupt is generated. The compare match is saved by hardware, and IT20UDC is cleared at the next count clock after the match.

The IT20CM1 register is a compare register used to set the PWM output duty. Set the duty required for the PWM cycle.

Figure 7-11. PWM Signal Output Example (When IT20ALVT Bit = 0)



Cautions 1. Changing the values of the IT20CM0 and IT20CM1 registers is prohibited during IT20UDC operation (IT20CE bit of IT20TMC register = 1).

2. Changing the value of the IT20ALVT bit of the IT20TUM register is prohibited during IT20UDC operation.

3. PWM signal output is performed from the second PWM cycle after the IT20CE bit is set (to 1).

7.4.3 Operation in UDC mode

(1) Overview of operation in UDC mode

The count clock input to IT20UDC in the UDC mode (IT20CMD bit of IT20TUM register = 1) can only be externally input from the TIT20IUD and TIT20CUD pins. Up/down count judgment in the UDC mode is determined based on the phase difference of the TIT20IUD and TIT20CUD pin inputs according to the IT20PRM register setting (there is a total of four choices).

Table 7-4. List of Count Operations in UDC Mode

IT20PRM Register			Operation Mode	IT20UDC Operation
IT20PRM2	IT20PRM1	IT20PRM0		
1	0	0	Mode 1	Down count when TIT20CUD = high level Up count when TIT20CUD = low level
1	0	1	Mode 2	Up count upon detection of valid edge of TIT20IUD input Down count upon detection of valid edge of TIT20CUD input
1	1	0	Mode 3	Automatic judgment by TIT20CUD input level upon detection of valid edge of TIT20IUD input
1	1	1	Mode 4	Automatic judgment upon detection of both edges of TIT20IUD input and both edges of TIT20CUD input

The UDC mode is further divided into two modes according to the IT20UDC clear conditions (a count operation is performed only with TIT20IUD and TIT20CUD input in both modes).

- **UDC mode A (IT20TUM register's IT20CMD bit = 1, IT20MSEL bit = 0)**
  The IT20UDC clear source can be selected as only external clear input (TIT20CLR), a match signal between the IT20UDC count value and the IT20CM0 set value during up count operation, or the logical sum (OR) of the two signals, using bits IT20CLR1 and IT20CLR0 of the IT20TMC register. IT20UDC can transfer the value of IT20CM0 upon occurrence of an IT20UDC underflow.

- **UDC mode B (IT20TUM register's IT20CMD bit = 1, IT20MSEL bit = 1)**
  The status of IT20UDC after a match of the IT20UDC count value and IT20CM0 set value is as follows.
  <1> In the case of an up count operation, IT20UDC is cleared (0000H), and the INTCM10 interrupt is generated.
  <2> In the case of a down count operation, the IT20UDC count value is decremented (–1).

  The status of IT20UDC after a match of the IT20UDC count value and IT20CM1 set value is as follows.
  <1> In the case of an up count operation, the IT20UDC count value is incremented (+1).
  <2> In the case of a down count operation, IT20UDC is cleared (0000H), and the INTCM11 interrupt is generated.

(2) Up/down count operation in UDC mode

IT20UDC up/down count judgment in the UDC mode is determined based on the phase difference of the TIT20IUD and TIT20CUD pin inputs according to the IT20PRM register setting.

(a) Mode 1 (IT20PRM2 bit = 1, IT20PRM1 bit = 0, IT20PRM0 bit = 0)

In mode 1, the following count operations are performed based on the level of the TIT20CUD pin upon detection of the valid edge of the TIT20IUD pin.

- IT20UDC down count operation when TIT20CUD pin = high level
- IT20UDC up count operation when TIT20CUD pin = low level

Figure 7-12. Mode 1 (When Rising Edge Is Specified as Valid Edge of TIT20IUD Pin)

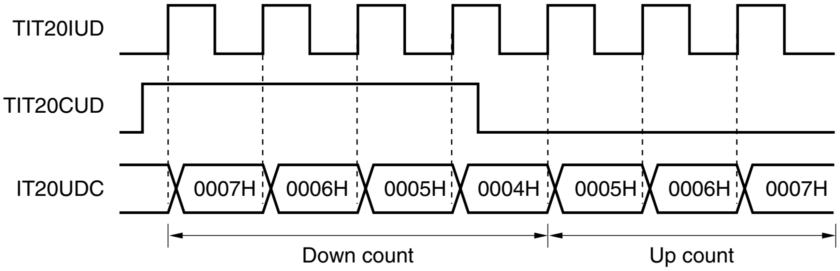
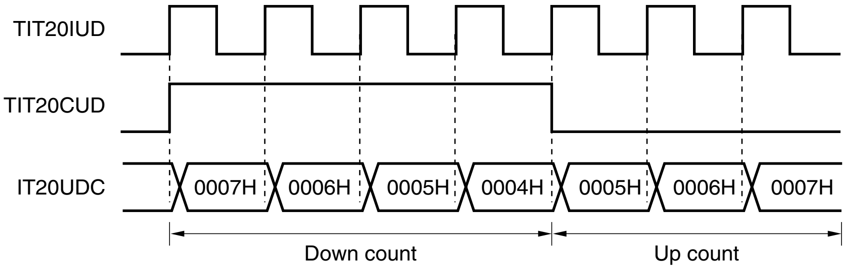


Figure 7-13. Mode 1 (When Rising Edge Is Specified as Valid Edge of TIT20IUD Pin): In Case of Simultaneous TIT20IUD, TIT20CUD Pin Edge Timing



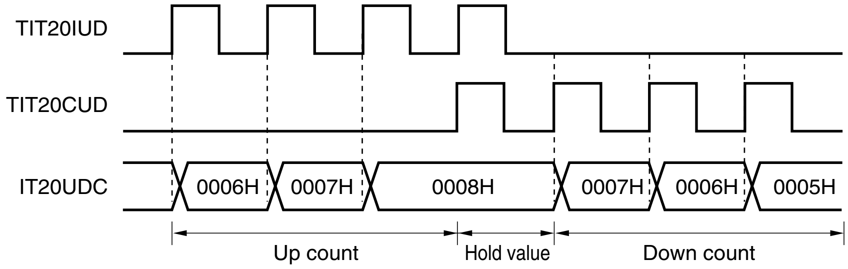
(b) Mode 2 (IT20PRM2 bit = 1, IT20PRM1 bit = 0, IT20PRM0 bit = 1)

The count conditions in mode 2 are as follows.

- IT20UDC up count upon detection of valid edge of TIT20IUD pin
- IT20UDC down count upon detection of valid edge of TIT20CUD pin

Caution If the count clock is simultaneously input to the TIT20IUD pin and the TIT20CUD pin, a count operation is not performed and the immediately preceding value is held.

Figure 7-14. Mode 2 (When Rising Edge Is Specified as Valid Edge of TIT20IUD, TIT20CUD Pins)



(c) Mode 3 (IT20PRM2 = 1, IT20PRM1 = 1, IT20PRM0 = 0)

In mode 3, when two signals 90 degrees out of phase are input to the TIT20IUD and TIT20CUD pins, the level of the TIT20CUD pin is sampled at the input of the valid edge of the TIT20IUD pin (see **Figure 7-15**).

If the TIT20CUD pin level sampled at the valid edge input to the TIT20IUD pin is low, IT20UDC counts down when the valid edge is input to the TIT20IUD pin.

If the TIT20CUD pin level sampled at the valid edge input to the TIT20IUD pin is high, IT20UDC counts up when the valid edge is input to the TIT20IUD pin.

Figure 7-15. Mode 3 (When Rising Edge Is Specified as Valid Edge of TIT20IUD pin)

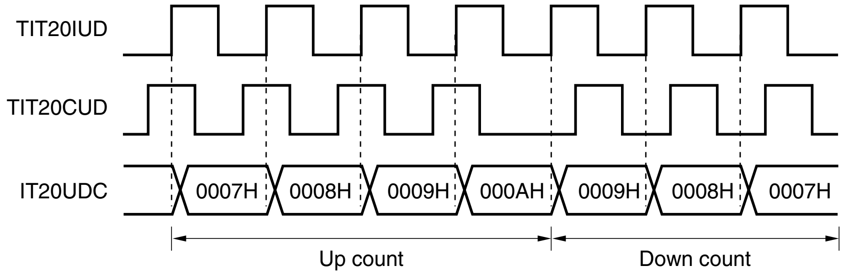
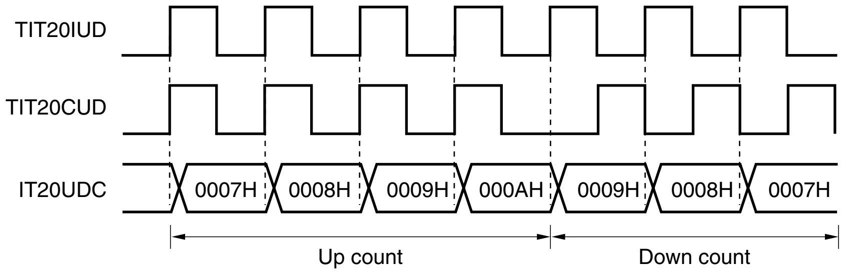


Figure 7-16. Mode 3 (When Rising Edge Is Specified as Valid Edge of TIT20IUD Pin): In Case of Simultaneous TIT20IUD, TIT20CUD Pin Edge Timing

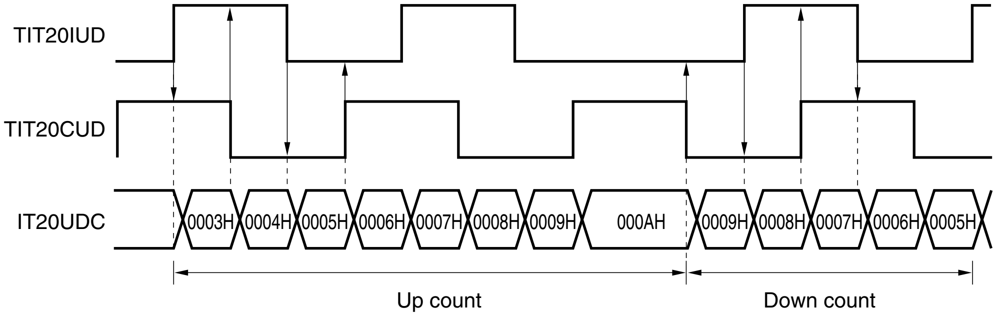


(d) Mode 4 (IT20PRM2 = 1, IT20PRM1 = 1, IT20PRM0 = 1)

In mode 4, when two signals out of phase are input to the TIT20IUD and TIT20CUD pins, the up/down operation is automatically judged and counting is performed according to the timing shown in **Figure 7-17**.

In mode 4, counting is executed at both the rising and falling edges of the two signals input to the TIT20IUD and TIT20CUD pins. Therefore, IT20UDC counts four times per cycle of an input signal (×4 count).

Figure 7-17. Mode 4



Cautions 1. When mode 4 is specified as the operation mode of IT20UDC, the valid edge specifications for the TIT20IUD and TIT20CUD pins are not valid.

2. If the TIT20IUD pin edge and TIT20CUD pin edge are input simultaneously in mode 4, IT20UDC continues the same count operation (up or down) it was performing immediately before the input.

(3) Operation in UDC mode A

(a) Interval operation

The operations at the count clock following a match of the IT20UDC count value and the IT20CM0 set value are as follows.

- In case of up count operation: IT20UDC is cleared (0000H) and the INTCM10 interrupt is generated.
- In case of down count operation: The IT20UDC count value is decremented (–1) and the INTCM10 interrupt is generated.

Remark The interval operation can be combined with the transfer operation.

(b) Transfer operation

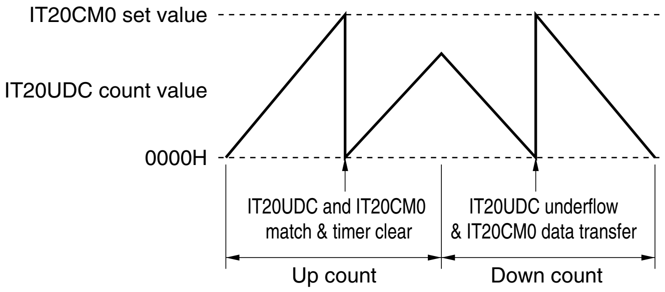
The operations at the next count clock after the count value of IT20UDC becomes 0000H during an IT20UDC count down operation are as follows.

- In case of down count operation: The data held in IT20CM0 is transferred.
- In case of up count operation: The IT20UDC count value is incremented (+1).

Remarks 1. Transfer enable/disable can be set using the IT20RLEN bit of the IT20TMC register.

2. The transfer operation can be combined with the interval operation.

Figure 7-18. Example of IT20UDC Operation When Interval Operation and Transfer Operation Are Combined


IT20CM0 set value
IT20UDC count value
0000H
IT20UDC and IT20CM0 match & timer clear
IT20UDC underflow & IT20CM0 data transfer
Up count
Down count

(c) Compare function

IT20UDC connects two compare register (IT20CM0, IT20CM1) channels and two capture/compare register (IT20CC0, IT20CC1) channels.

When the IT20UDC count value and the set value of one of the compare registers match, a match interrupt (INTCM10, INTCM11, INTCC10[Note], INTCC11[Note]) is output.

Note This match interrupt is generated when IT20CC0 and IT20CC1 are set to the compare register mode.

(d) Capture function

IT20UDC connects two capture/compare register (IT20CC0, IT20CC1) channels.

When IT20CC0 and IT20CC1 are set to the capture register mode, the value of IT20UDC is captured in synchronization with the corresponding capture trigger signal.

When IT20CC0 and IT20CC1 are set to the capture register mode, a capture interrupt (INTCC10, INTCC11) is generated upon detection of the valid edge.

(4) Operation in UDC mode B

(a) Basic operation

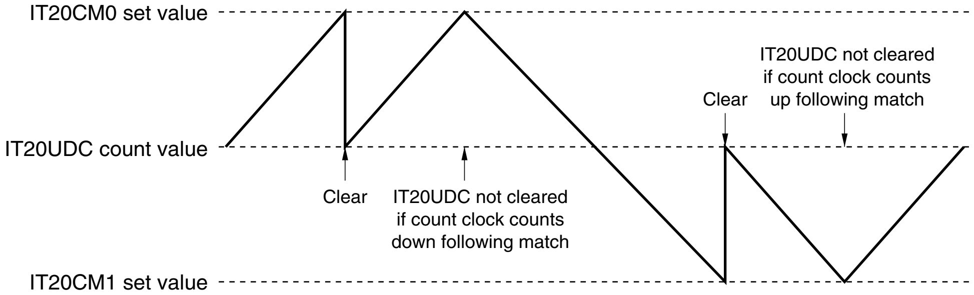
The operations at the next count clock after the count value of IT20UDC and the IT20CM0 set value match when IT20UDC is in UDC mode B are as follows.

- In case of up count operation: IT20UDC is cleared (0000H) and the INTCM10 interrupt is generated.
- In case of down count operation: The IT20UDC count value is decremented (–1).

The operations at the next count clock after the count value of IT20UDC and the IT20CM1 set value match when IT20UDC is in UDC mode B are as follows.

- In case of up count operation: The IT20UDC count value is incremented (+1).
- In case of down count operation: IT20UDC is cleared (0000H) and the INTCM11 interrupt is generated.

Figure 7-19. Example of IT20UDC Operation in UDC Mode


IT20CM0 set value
IT20UDC not cleared if count clock counts up following match
Clear
IT20UDC count value
Clear
IT20UDC not cleared if count clock counts down following match
IT20CM1 set value

(b) Compare function

IT20UDC connects two compare register (IT20CM0, IT20CM1) channels and two capture/compare register (IT20CC0, IT20CC1) channels.

When the IT20UDC count value and the set value of one of the compare registers match, a match interrupt (INTCM10 (only during up count operation), INTCM11 (only during down count operation), INTCC10[Note], INTCC11[Note]) is output.

Note This match interrupt is generated when IT20CC0 and IT20CC1 are set to the compare register mode.

(c) Capture function

IT20UDC connects two capture/compare register (IT20CC0, IT20CC1) channels.

When IT20CC0 and IT20CC1 are set to the capture register mode, the value of IT20UDC is captured in synchronization with the corresponding capture trigger signal.

When IT20CC0 and IT20CC1 are set to the capture register mode, a capture interrupt (INTCC10, INTCC11) is generated upon detection of the valid edge.

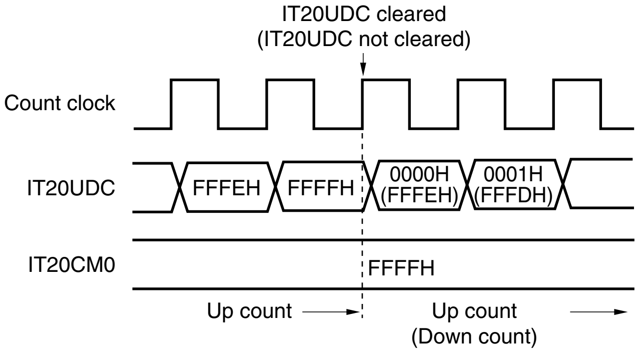
7.5 Internal Operation of 16-Bit Up/down Counter ITENC20

7.5.1 Clearing of count value in UDC mode B

When IT20UDC is in UDC mode B, the count value clear operation is as follows.

- In case of IT20UDC up count operation: IT20UDC is cleared upon match with IT20CM0
- In case of IT20UDC down count operation: IT20UDC is cleared upon match with IT20CM1

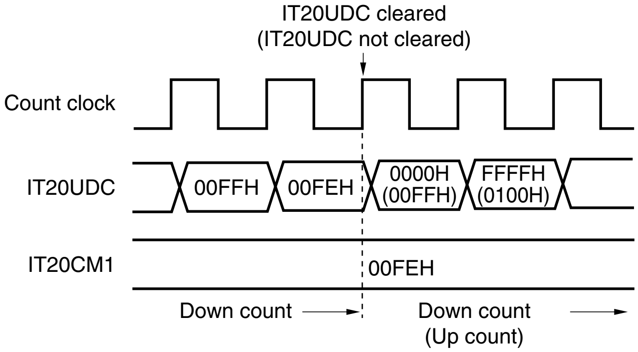
Figure 7-20. Clear Operation upon Match with IT20CM0 during IT20UDC Up Count Operation



Remarks 1. Rising edge of count clock set as valid edge.

2. The items in parentheses in the above figure apply to down count operations.

Figure 7-21. Clear Operation upon Match with IT20CM1 during IT20UDC Down Count Operation



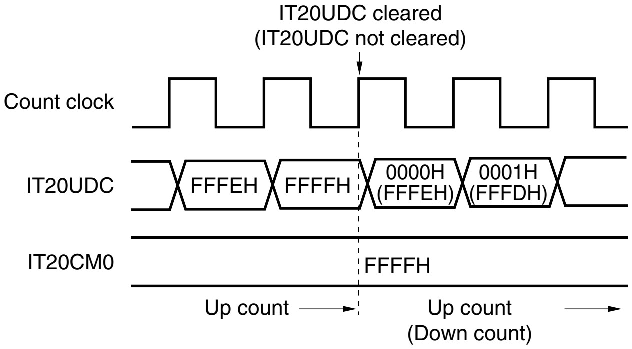
Remarks 1. Rising edge of count clock set as valid edge.

2. The items in parentheses in the above figure apply to up count operations.

7.5.2 Clearing of count value upon occurrence of compare match

The internal operation during an IT20UDC clear operation upon occurrence of a compare match is as follows.

Figure 7-22. Count Value Clear Operation upon Compare Match



Caution The operations at the next count clock after the count value of IT20UDC and the IT20CM0 set value match are as follows.

- **In case of up count: Clear operation is performed.**
- **In case of down count: Clear operation is not performed.**

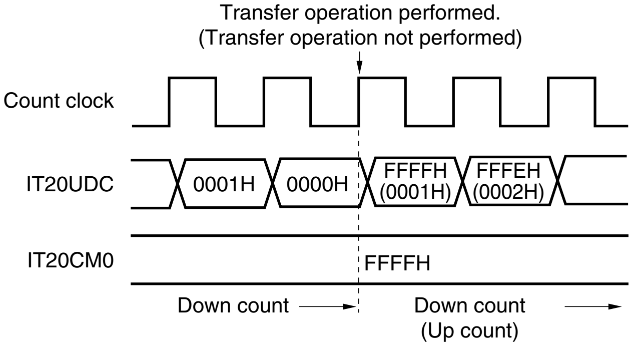
Remarks 1. Rising edge of count clock set as valid edge.

2. The items in parentheses in the above figure apply to down count operations.

7.5.3 Transfer operation

The internal operation during IT20UDC transfer operation is as follows.

Figure 7-23. Internal Operation During Transfer Operation



Caution The count operations after the IT20UDC count value becomes 0000H are as follows.

- **In case of down count: Transfer operation is performed.**
- **In case of up count: Transfer operation is not performed.**

Remarks 1. Rising edge of count clock set as valid edge.

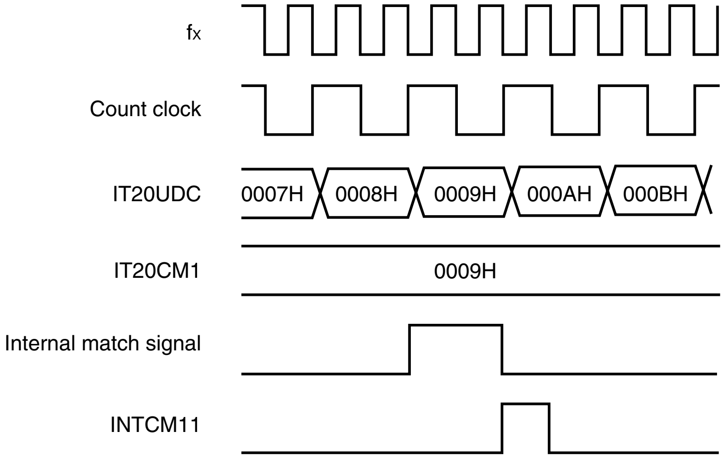
2. The items in parentheses in the above figure apply to up count operations.

7.5.4 Interrupt signal output upon compare match

An interrupt signal is output when the count value of IT20UDC matches the set value of the IT20CM0, IT20CM1, IT20CC0[Note], or IT20CC1[Note] register. The interrupt generation timing is as follows.

Note When IT20CC0 and IT20CC1 are set to the compare register mode.

Figure 7-24. Interrupt Output upon Compare Match
(IT20CM1 with Operation Mode Set to General-Purpose Timer Mode and Count Clock Set to fx/2)



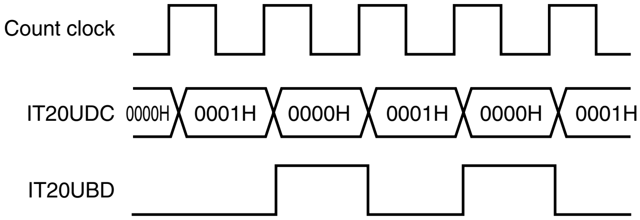
Remark fx: Internal system clock

An interrupt signal such as the one illustrated in Figure 7-24 is output at the next count following a match of the IT20UDC count value and the set value of the corresponding compare register.

7.5.5 IT20UBD flag (bit 0 of IT20STS register) operation

In the UDC mode (IT20CMD bit of IT20TUM register = 1), the IT20UBD flag changes as follows during an IT20UDC up/down count operation at every internal operation clock.

Figure 7-25. IT20UBD Flag Operation



CHAPTER 8 16-BIT TIMER/EVENT COUNTER 00

8.1 Functions of 16-Bit Timer/Event Counter 00

16-bit timer/event counter 00 has the following functions.

- Interval timer
- PPG output
- Pulse width measurement
- External event counter
- Square-wave output
- One-shot pulse output

(1) Interval timer

16-bit timer/event counter 00 generates an interrupt request at the preset time interval.

(2) PPG output

16-bit timer/event counter 00 can output a rectangular wave whose frequency and output pulse width can be set freely.

(3) Pulse width measurement

16-bit timer/event counter 00 can measure the pulse width of an externally input signal.

(4) External event counter

16-bit timer/event counter 00 can measure the number of pulses of an externally input signal.

(5) Square-wave output

16-bit timer/event counter 00 can output a square wave with any selected frequency.

(6) One-shot pulse output

16-bit timer/event counter 00 can output a one-shot pulse whose output pulse width can be set freely.

8.2 Configuration of 16-Bit Timer/Event Counter 00

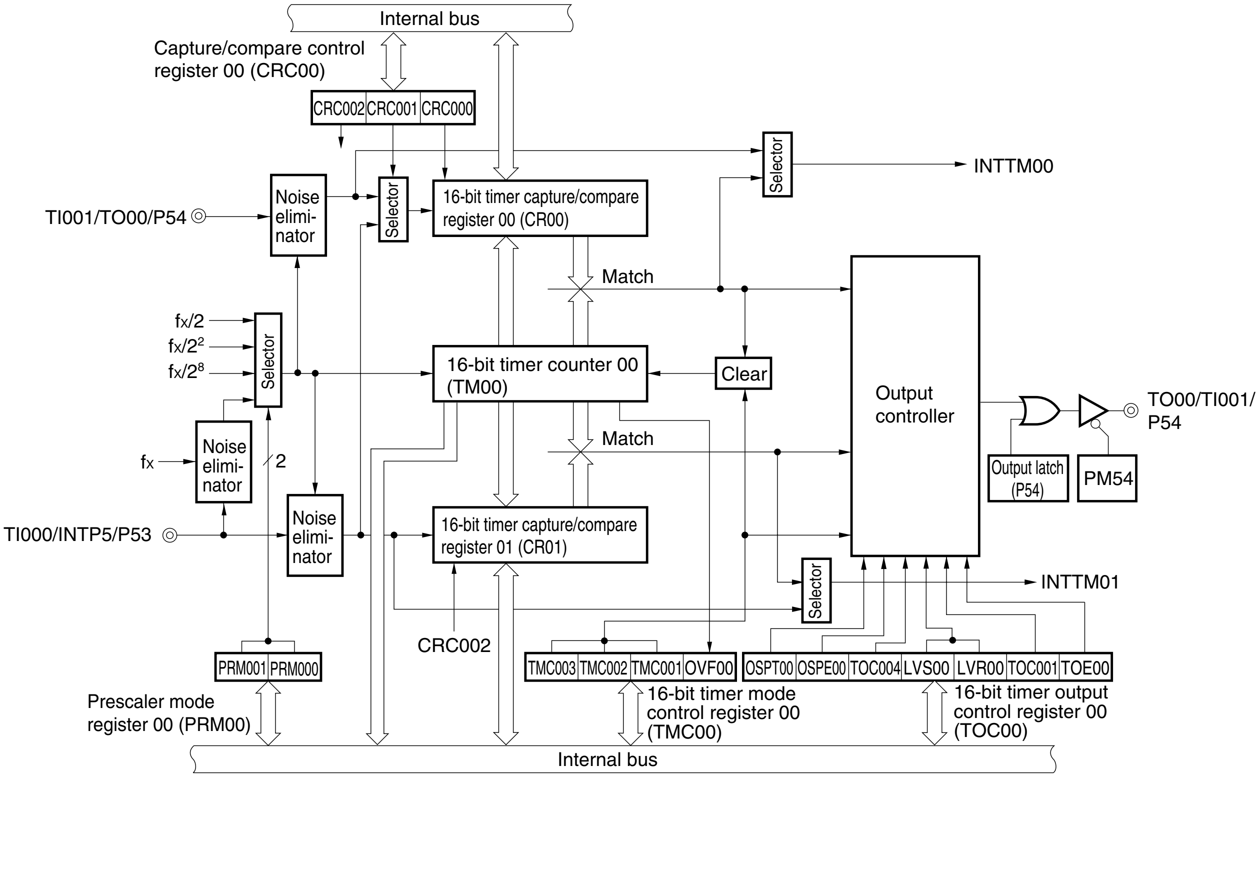
16-bit timer/event counter 00 consists of the following hardware.

Table 8-1. Configuration of 16-Bit Timer/Event Counter 00

Item	Configuration
Timer counter	16 bits (TM00)
Register	16-bit timer capture/compare register: 16 bits (CR00, CR01)
Timer input	TI000, TI001
Timer output	TO00, output controller
Control registers	16-bit timer mode control register 00 (TMC00) 16-bit timer capture/compare control register 00 (CRC00) 16-bit timer output control register 00 (TOC00) Prescaler mode register 00 (PRM00) Port mode register 5 (PM5) Port register 5 (P5)

Figures 8-1 shows the block diagrams.

Figure 8-1. Block Diagram of 16-Bit Timer/Event Counter 00



(1) 16-bit timer counter 00 (TM00)

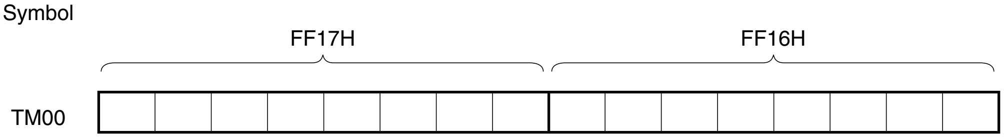
TM00 is a 16-bit read-only register that counts count pulses.

The counter is incremented in synchronization with the rising edge of the input clock.

Figure 8-2. Format of 16-Bit Timer Counter 00 (TM00)

Address: FF16H, FF17H After reset: 0000H R

Symbol



The count value is reset to 0000H in the following cases.

<1> At $\overline{\text{RESET}}$ input

<2> If TMC003 and TMC002 are cleared

<3> If the valid edge of the TI000 pin is input in the mode in which clear & start occurs upon input of the valid edge of the TI000 pin

<4> If TM00 and CR00 match in the mode in which clear & start occurs on a match of TM00 and CR00

<5> OSPT00 is set in one-shot pulse output mode

(2) 16-bit timer capture/compare register 00 (CR00)

CR00 is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or as a compare register is set by bit 0 (CRC000) of capture/compare control register 00 (CRC00).

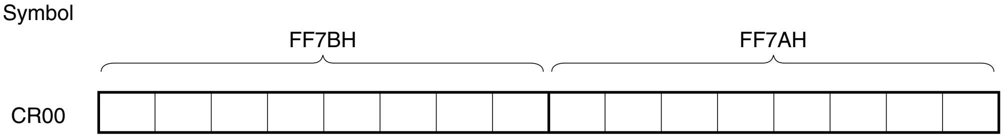
CR00 can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 8-3. Format of 16-Bit Timer Capture/Compare Register 00 (CR00)

Address: FF7AH, FF7BH After reset: 0000H R/W

Symbol



- **When CR00 is used as a compare register**

  The value set in CR00 is constantly compared with 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM00) is generated if they match. The set value is held until CR00 is rewritten.

- **When CR00 is used as a capture register**

  It is possible to select the valid edge of the TI000 pin or the TI001 pin as the capture trigger. The TI000 or TI001 pin valid edge is set using prescaler mode register 00 (PRM00) (see **Table 8-2**).

Table 8-2. CR00 Capture Trigger and Valid Edges of TI000 and TI001 Pins

(1) TI000 pin valid edge selected as capture trigger (CRC001 = 1, CRC000 = 1)

CR00 Capture Trigger	TI000 Pin Valid Edge		
		ES001	ES000
Falling edge	Rising edge	0	1
Rising edge	Falling edge	0	0
No capture operation	Both rising and falling edges	1	1

(2) TI001 pin valid edge selected as capture trigger (CRC001 = 0, CRC000 = 1)

CR00 Capture Trigger	TI001 Pin Valid Edge		
		ES101	ES100
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Remarks 1. Setting ES001, ES000 = 1, 0 and ES101, ES100 = 1, 0 is prohibited.

2. ES001, ES000: Bits 5 and 4 of prescaler mode register 00 (PRM00)
ES101, ES100: Bits 7 and 6 of prescaler mode register 00 (PRM00)
CRC001, CRC000: Bits 1 and 0 of capture/compare control register 00 (CRC00)

Cautions 1. Set a value other than 0000H in CR00 in the mode in which clear & start occurs on a match of TM00 and CR00.

2. In the free-running mode and in the clear mode using the valid edge of the TI000 pin, if CR00 is cleared to 0000H, an interrupt request (INTTM00) is generated when the value of CR00 changes from 0000H to 0001H following overflow (FFFFH). INTTM00 is generated after TM00 and CR00 match, after the valid edge of the TI000 pin is detected, or after the timer is cleared by a one-shot trigger.

3. When P54 is used as the valid edge input of the TI001 pin, it cannot be used as the timer output (TO00). Moreover, when P54 is used as TO00, it cannot be used as the valid edge input of the TI001 pin.

4. When CR00 is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

5. Do not rewrite CR00 during TM00 operation.

(3) 16-bit timer capture/compare register 01 (CR01)

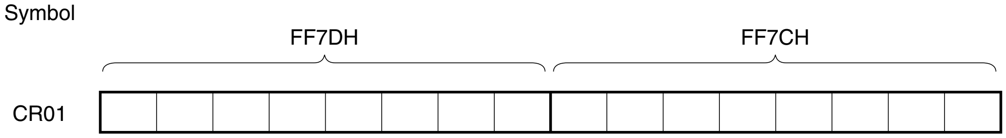
CR01 is a 16-bit register that has the functions of both a capture register and a compare register. Whether it is used as a capture register or a compare register is set by bit 2 (CRC002) of capture/compare control register 00 (CRC00).

CR01 can be set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 8-4. Format of 16-Bit Timer Capture/Compare Register 01 (CR01)

Address: FF7CH, FF7DH After reset: 0000H R/W



- **When CR01 is used as a compare register**

  The value set in the CR01 is constantly compared with 16-bit timer counter 00 (TM00) count value, and an interrupt request (INTTM01) is generated if they match. The set value is held until CR01 is rewritten.

- **When CR01 is used as a capture register**

  It is possible to select the valid edge of the TI000 pin as the capture trigger. The TI000 pin valid edge is set by prescaler mode register 00 (PRM00) (see **Table 8-3**).

Table 8-3. CR01 Capture Trigger and Valid Edge of TI000 Pin (CRC002 = 1)

CR01 Capture Trigger	TI000 Pin Valid Edge		
		ES001	ES000
Falling edge	Falling edge	0	0
Rising edge	Rising edge	0	1
Both rising and falling edges	Both rising and falling edges	1	1

Remarks 1. Setting ES001, ES000 = 1, 0 is prohibited.

2. ES001, ES000: Bits 5 and 4 of prescaler mode register 00 (PRM00)
CRC002: Bit 2 of capture/compare control register 00 (CRC00)

Cautions 1. If the CR01 register is cleared to 0000H, an interrupt request (INTTM01) is generated after the TM00 register overflows, after the timer is cleared and started on a match between the TM00 register and the CR00 register, or after the timer is cleared by the valid edge of the TI000 pin or a one-shot trigger.

2. When CR01 is used as a capture register, read data is undefined if the register read time and capture trigger input conflict (the capture data itself is the correct value).
If count stop input and capture trigger input conflict, the captured data is undefined.

3. CR01 can be rewritten during TM00 operation. For the details of how to rewrite CR01, see Caution 2 of Figure 8-15.

8.3 Registers Controlling 16-Bit Timer/Event Counter 00

The following six registers are used to control 16-bit timer/event counter 00.

- 16-bit timer mode control register 00 (TMC00)
- Capture/compare control register 00 (CRC00)
- 16-bit timer output control register 00 (TOC00)
- Prescaler mode register 00 (PRM00)
- Port mode register 5 (PM5)
- Port register 5 (P5)

(1) 16-bit timer mode control register 00 (TMC00)

This register sets the 16-bit timer operating mode, 16-bit timer counter 00 (TM00) clear mode, and output timing, and detects an overflow.

TMC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TMC00 to 00H.

Caution 16-bit timer counter 00 (TM00) starts operation at the moment TMC002 and TMC003 are set to values other than 0, 0 (operation stop mode), respectively. Clear TMC002 and TMC003 to 0, 0 to stop the operation.

Figure 8-5. Format of 16-Bit Timer Mode Control Register 00 (TMC00)

Address: FF7EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
TMC00	0	0	0	0	TMC003	TMC002	TMC001	OVF00

TMC003	TMC002	TMC001	Operating mode and clear mode selection	TO00 inversion timing selection	Interrupt request generation
0	0	0	Operation stop (TM00 cleared to 0)	No change	Not generated
0	0	1			
0	1	0	Free-running mode	Match between TM00 and CR00 or match between TM00 and CR01	Generated on match between TM00 and CR00, or match between TM00 and CR01
0	1	1		Match between TM00 and CR00, match between TM00 and CR01 or TI000 pin valid edge	
1	0	0	Clear & start occurs on TI000 pin valid edge	–	
1	0	1			
1	1	0	Clear & start occurs on match between TM00 and CR00	Match between TM00 and CR00 or match between TM00 and CR01	
1	1	1		Match between TM00 and CR00, match between TM00 and CR01 or TI000 pin valid edge	

OVF00	16-bit timer counter 00 (TM00) overflow detection
0	Overflow not detected
1	Overflow detected

Cautions 1. Timer operation must be stopped before writing to bits other than the OVF00 flag.

2. Set the valid edge of the TI000/P53 pin using prescaler mode register 00 (PRM00).

3. If any of the following modes is selected: the mode in which clear & start occurs on match between TM00 and CR00, the mode in which clear & start occurs at the TI000 pin valid edge, or free-running mode, when the set value of CR00 is FFFFH and the TM00 value changes from FFFFH to 0000H, the OVF00 flag is set to 1.

Remarks 1. TO00: 16-bit timer/event counter 00 output pin

2. TI000: 16-bit timer/event counter 00 input pin

3. TM00: 16-bit timer counter 00

4. CR00: 16-bit timer capture/compare register 00

5. CR01: 16-bit timer capture/compare register 01

(2) Capture/compare control register 00 (CRC00)

This register controls the operation of the 16-bit timer capture/compare registers (CR00, CR01).

CRC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CRC00 to 00H.

Figure 8-6. Format of Capture/Compare Control Register 00 (CRC00)

Address: FF6AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CRC00	0	0	0	0	0	CRC002	CRC001	CRC000

CRC002	CR01 operating mode selection
0	Operates as compare register
1	Operates as capture register

CRC001	CR00 capture trigger selection
0	Captures on valid edge of TI001 pin
1	Captures on valid edge of TI000 pin by reverse phase

CRC000	CR00 operating mode selection
0	Operates as compare register
1	Operates as capture register

Cautions 1. Timer operation must be stopped before setting CRC00.

2. When the mode in which clear & start occurs on a match between TM00 and CR00 is selected with 16-bit timer mode control register 00 (TMC00), CR00 should not be specified as a capture register.

3. The capture operation is not performed if both the rising and falling edges are specified as the valid edge of the TI000 pin.

4. To ensure that the capture operation is performed properly, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

(3) 16-bit timer output control register 00 (TOC00)

This register controls the operation of 16-bit timer/event counter 00 output controller. It sets/resets the timer output F/F, enables/disables output inversion and 16-bit timer/event counter 00 timer output, enables/disables the one-shot pulse output operation, and sets the one-shot pulse output trigger via software.

TOC00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears TOC00 to 00H.

Figure 8-7. Format of 16-Bit Timer Output Control Register 00 (TOC00)

Address: FF6BH After reset: 00H R/W

Symbol	7	<6>	<5>	4	<3>	<2>	1	<0>
TOC00	0	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00

OSPT00	One-shot pulse output trigger control via software
0	No one-shot pulse trigger
1	One-shot pulse trigger

OSPE00	One-shot pulse output operation control
0	Successive pulse output mode
1	One-shot pulse output mode[Note]

TOC004	Timer output F/F control using match of CR01 and TM00
0	Disables inversion operation
1	Enables inversion operation

LVS00	LVR00	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TOC001	Timer output F/F control using match of CR00 and TM00
0	Disables inversion operation
1	Enables inversion operation

TOE00	Timer output control
0	Disables output (output fixed to level 0)
1	Enables output

Note The one-shot pulse output mode operates correctly only in the free-running mode and the mode in which clear & start occurs at the TI000 pin valid edge. In the mode in which clear & start occurs on a match between the TM00 register and CR00 register, one-shot pulse output is not possible because an overflow does not occur.

Cautions

1. **Timer operation must be stopped before setting other than TOC004.**
2. **If LVS00 and LVR00 are read, 0 is read.**
3. **OSPT00 is automatically cleared after data is set, so 0 is read.**
4. **Do not set OSPT00 to 1 other than in one-shot pulse output mode.**
5. **A write interval of two cycles or more of the count clock selected by prescaler mode register 00 (PRM00) is required to write to OSPT00 successively.**
6. **Do not set LVS00 to 1 before TOE00, and do not set LVS00 and TOE00 to 1 simultaneously.**

(4) Prescaler mode register 00 (PRM00)

This register is used to set the 16-bit timer counter 00 (TM00) count clock and valid edges of the TI000 and TI001 pin inputs.

PRM00 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears PRM00 to 00H.

Figure 8-8. Format of Prescaler Mode Register 00 (PRM00)

Address: FF7FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PRM00	ES101	ES100	ES001	ES000	0	0	PRM001	PRM000

ES101	ES100	TI001 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

ES001	ES000	TI000 pin valid edge selection
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both falling and rising edges

PRM001	PRM000	Count clock selection[Note 1]
0	0	$f_X/2$ (10 MHz)
0	1	$f_X/2^2$ (5 MHz)
1	0	$f_X/2^8$ (78.125 kHz)
1	1	TI000 pin valid edge[Note 2]

Notes 1. Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz

2. The external clock requires a pulse two cycles longer than internal clock (f_X).

Remarks 1. f_X: X1 input clock oscillation frequency

2. TI000, TI001: 16-bit timer/event counter 00 input pin

3. Figures in parentheses are for operation with f_X = 20 MHz.

Cautions 1. When the internal oscillation clock is selected as the source clock to the CPU, the clock of the internal oscillator is divided and supplied as the count clock. If the count clock is the internal oscillation clock, the operation of 16-bit timer/event counter 00 is not guaranteed. When an external clock is used and when the internal oscillation clock is selected and supplied to the CPU, the operation of 16-bit timer/event counter 00 is not guaranteed, either, because the internal oscillation clock is supplied as the sampling clock to eliminate noise.

2. Always set data to PRM00 after stopping the timer operation.

3. If the valid edge of the TI000 pin is to be set for the count clock, do not set the clear & start mode using the valid edge of the TI000 pin and the capture trigger.

4. If the TI000 or TI001 pin is high level immediately after system reset, the rising edge is immediately detected after the rising edge or both the rising and falling edges are set as the valid edge(s) of the TI000 pin or TI001 pin to enable the operation of 16-bit timer counter 00 (TM00). Care is therefore required when pulling up the TI000 or TI001 pin. However, when re-enabling operation after the operation has been stopped once, the rising edge is not detected.

5. When P54 is used as the TI001 pin valid edge, it cannot be used as the timer output (TO00), and when used as TO00, it cannot be used as the TI001 pin valid edge.

(5) Port mode register 5 (PM5)

This register sets port 5 input/output in 1-bit units.

When using the P54/TO00/TI001 pin for timer output, clear PM54 and the output latch of P54 to 0.

When using the P54/TO00/TI001 and P53/TI000/INTP5 pins for timer input, clear PM54 and PM53 to 1. At this time, the output latches of P54 and P53 may be 0 or 1.

PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM5 to FFH.

Figure 8-9. Format of Port Mode Register 5 (PM5)

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	P5n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Operation of 16-Bit Timer/Event Counter 00

8.4.1 Interval timer operation

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 8-10 allows operation as an interval timer.

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figure 8-10** for the set value).
<2> Set any value to the CR00 register.
<3> Set the count clock by using the PRM00 register.
<4> Set the TMC00 register to start the operation (see **Figure 8-10** for the set value).

Caution CR00 cannot be rewritten during TM00 operation.

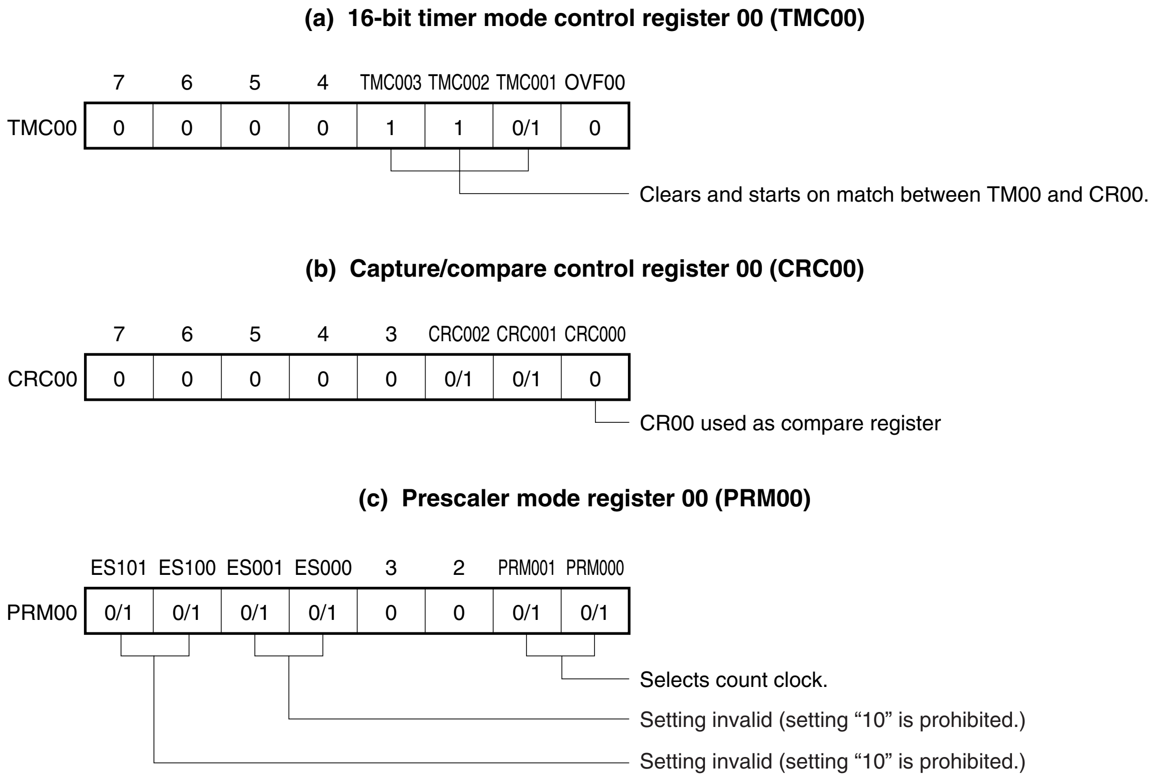
Remark For how to enable the INTTM00 interrupt, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

Interrupt requests are generated repeatedly using the count value preset in 16-bit timer capture/compare register 00 (CR00) as the interval.

When the count value of 16-bit timer counter 00 (TM00) matches the value set in CR00, counting continues with the TM00 value cleared to 0 and the interrupt request signal (INTTM00) is generated.

The count clock of 16-bit timer/event counter 00 can be selected with bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00).

Figure 8-10. Control Register Settings for Interval Timer Operation



(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0/1	0

Clears and starts on match between TM00 and CR00.

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0/1	0/1	0

CR00 used as compare register

(c) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0/1	0/1	0	0	0/1	0/1

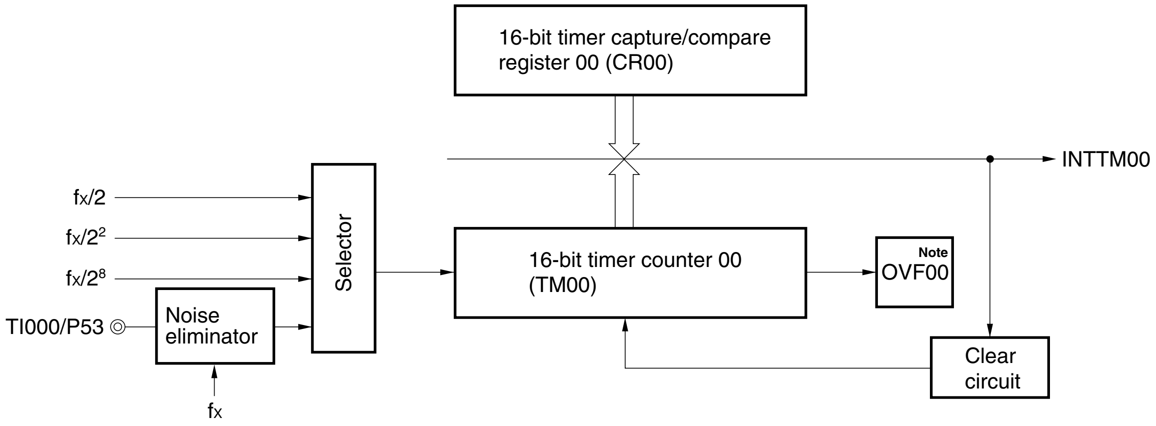
Selects count clock.

Setting invalid (setting “10” is prohibited.)

Setting invalid (setting “10” is prohibited.)

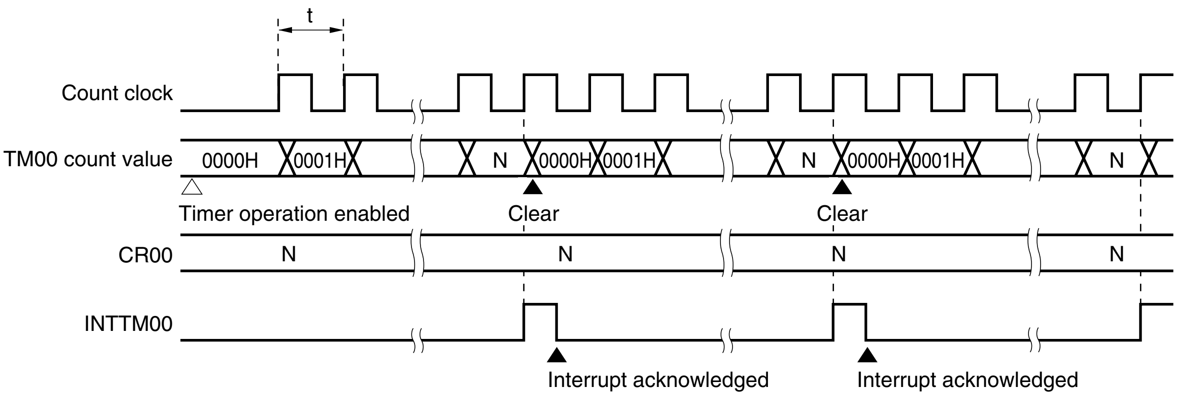
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the interval timer.
See the description of the respective control registers for details.

Figure 8-11. Interval Timer Configuration Diagram



Note OVF00 is set to 1 only when 16-bit timer capture/compare register 00 is set to FFFFH.

Figure 8-12. Timing of Interval Timer Operation



Remark Interval time = (N + 1) × t

N = 0001H to FFFFH

8.4.2 PPG output operations

Setting 16-bit timer mode control register 00 (TMC00) and capture/compare control register 00 (CRC00) as shown in Figure 8-13 allows operation as PPG (Programmable Pulse Generator) output.

Setting

The basic operation setting procedure is as follows.

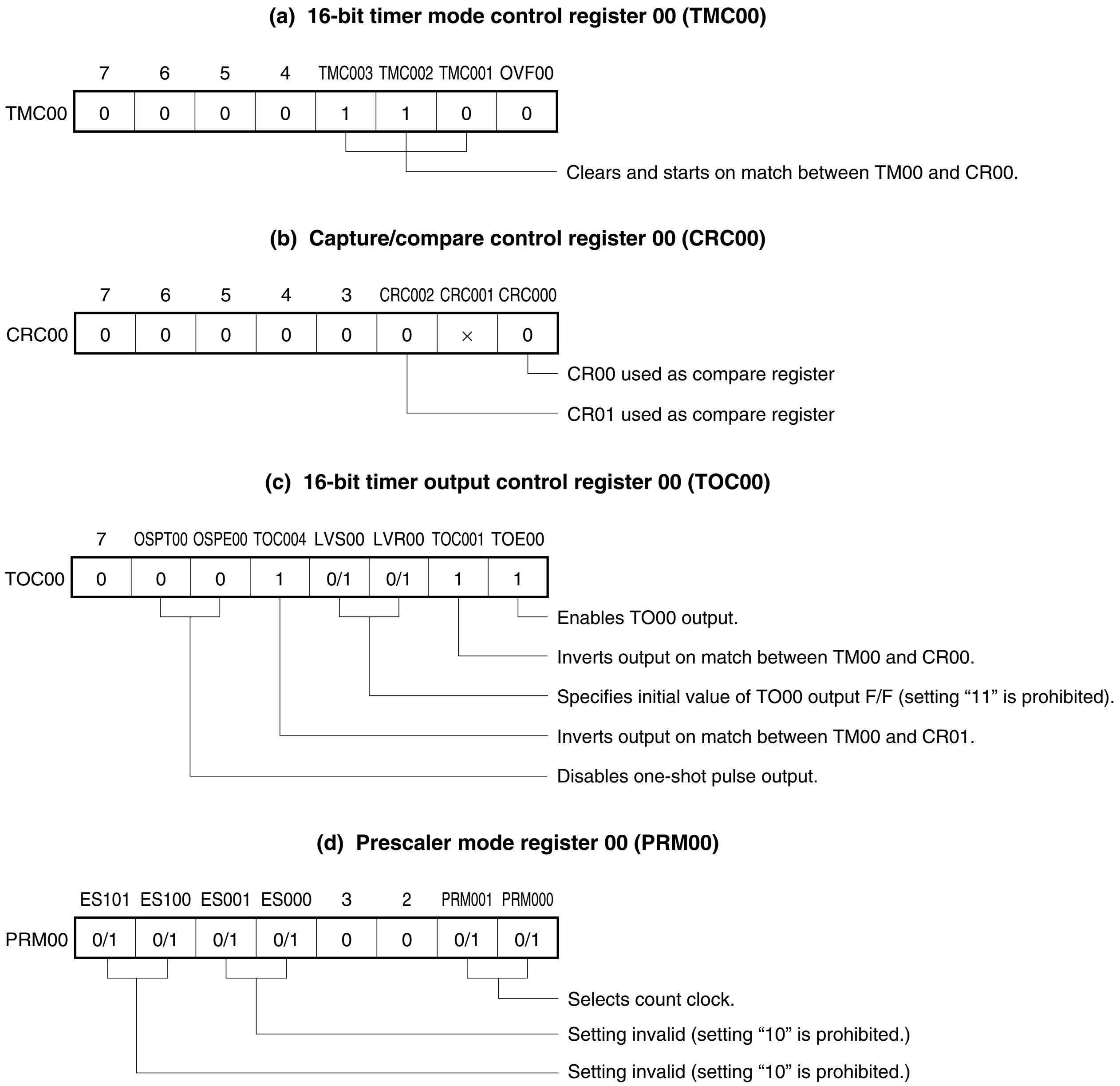
<1> Set the CRC00 register (see **Figure 8-13** for the set value).
<2> Set any value to the CR00 register as the cycle.
<3> Set any value to the CR01 register as the duty factor.
<4> Set the TOC00 register (see **Figure 8-13** for the set value).
<5> Set the count clock by using the PRM00 register.
<6> Set the TMC00 register to start the operation (see **Figure 8-13** for the set value).

Caution To change the value of the duty factor (the value of the CR01 register) during operation, see Caution 2 in Figure 8-15 PPG Output Operation Timing.

Remarks 1. For the setting of the TO00 pin, see **8.3 (5) Port mode register 5 (PM5)**.
2. For how to enable the INTTM00 interrupt, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

In the PPG output operation, rectangular waves are output from the TO00 pin with the pulse width and the cycle that correspond to the count values preset in 16-bit timer capture/compare register 01 (CR01) and in 16-bit timer capture/compare register 00 (CR00), respectively.

Figure 8-13. Control Register Settings for PPG Output Operation



(a) 16-bit timer mode control register 00 (TMC00)

(b) Capture/compare control register 00 (CRC00)

(c) 16-bit timer output control register 00 (TOC00)

(d) Prescaler mode register 00 (PRM00)

7 6 5 4 TMC003 TMC002 TMC001 OVF00
TMC00 0 0 0 0 1 1 0 0
Clears and starts on match between TM00 and CR00.
7 6 5 4 3 CRC002 CRC001 CRC000
CRC00 0 0 0 0 0 0 × 0
CR00 used as compare register
CR01 used as compare register
7 OSPT00 OSPE00 TOC004 LVS00 LVR00 TOC001 TOE00
TOC00 0 0 0 1 0/1 0/1 1 1
Enables TO00 output.
Inverts output on match between TM00 and CR00.
Specifies initial value of TO00 output F/F (setting “11” is prohibited).
Inverts output on match between TM00 and CR01.
Disables one-shot pulse output.
ES101 ES100 ES001 ES000 3 2 PRM001 PRM000
PRM00 0/1 0/1 0/1 0/1 0 0 0/1 0/1
Selects count clock.
Setting invalid (setting “10” is prohibited.)
Setting invalid (setting “10” is prohibited.)

Cautions 1. Values in the following range should be set in CR00 and CR01:
0000H ≤ CR01 < CR00 ≤ FFFFH

2. The cycle of the pulse generated through PPG output (CR00 setting value + 1) has a duty of (CR01 setting value + 1)/(CR00 setting value + 1).

Remark ×: Don’t care

Figure 8-14. Configuration Diagram of PPG Output

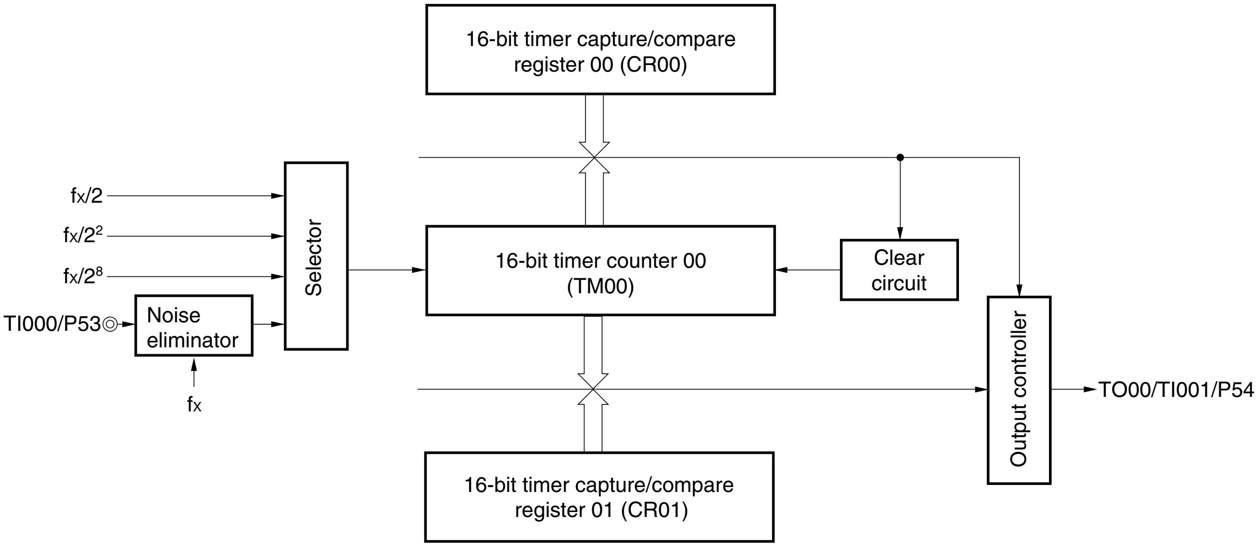
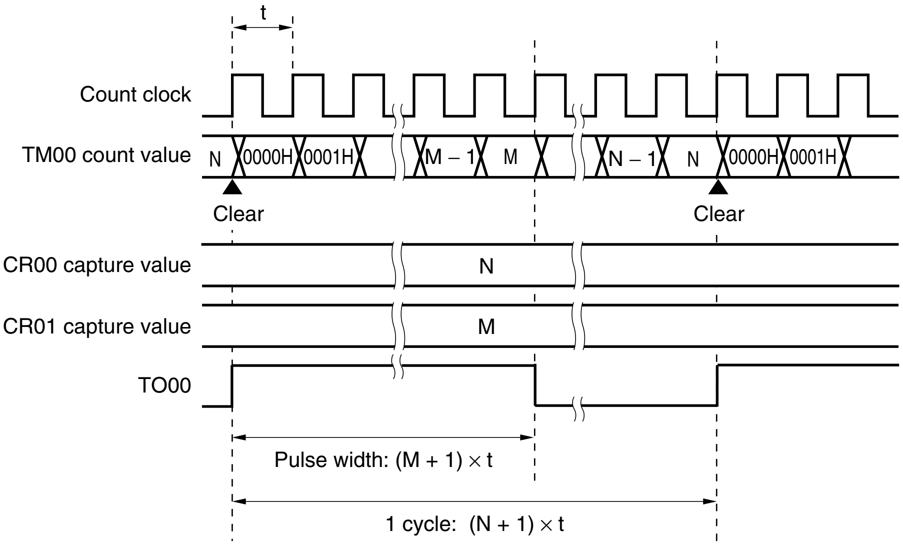

16-bit timer capture/compare register 00 (CR00)
fx/2
fx/2²
fx/2⁸
TI000/P53◎
Noise eliminator
fx
Selector
16-bit timer counter 00 (TM00)
Clear circuit
Output controller
TO00/TI001/P54
16-bit timer capture/compare register 01 (CR01)

Figure 8-15. PPG Output Operation Timing



Cautions 1. CR00 cannot be rewritten during TM00 operation.

2. In the PPG output operation, change the pulse width (rewrite CR01) during TM00 operation using the following procedure.

<1> Disable the timer output inversion operation by match of TM00 and CR01 (TOC004 = 0)

<2> Disable the INTTM01 interrupt (TMMK01 = 1)

<3> Rewrite CR01

<4> Wait for 1 cycle of the TM00 count clock

<5> Enable the timer output inversion operation by match of TM00 and CR01 (TOC004 = 1)

<6> Clear the interrupt request flag of INTTM01 (TMIF01 = 0)

<7> Enable the INTTM01 interrupt (TMMK01 = 0)

Remark 0000H ≤ M < N ≤ FFFFH

8.4.3 Pulse width measurement operations

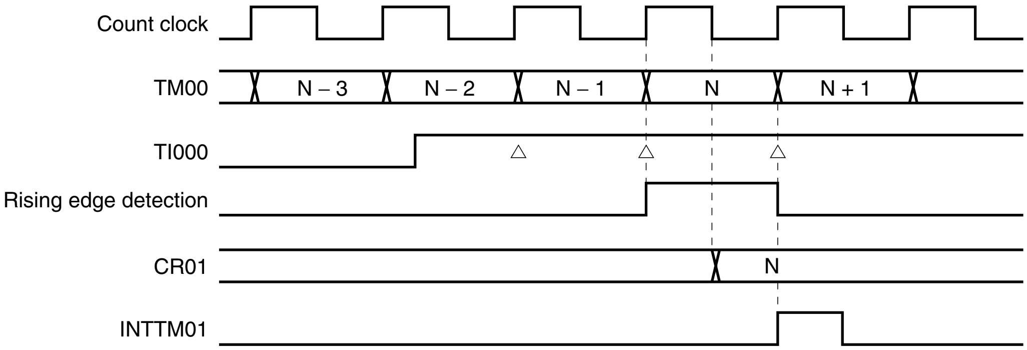
It is possible to measure the pulse width of the signals input to the TI000 pin and TI001 pin using 16-bit timer counter 00 (TM00).

There are two measurement methods: measuring with TM00 used in free-running mode, and measuring by restarting the timer in synchronization with the edge of the signal input to the TI000 pin.

When an interrupt occurs, read the valid value of the capture register, check the overflow flag, and then calculate the necessary pulse width. Clear the overflow flag after checking it.

The capture operation is not performed until the signal pulse width is sampled in the count clock cycle selected by prescaler mode register 00 (PRM00) and the valid level of the TI000 or TI001 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 8-16. CR01 Capture Operation with Rising Edge Specified



Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figures 8-17**, **8-20**, **8-22**, and **8-24** for the set value).
<2> Set the count clock by using the PRM00 register.
<3> Set the TMC00 register to start the operation (see **Figures 8-17**, **8-20**, **8-22**, and **8-24** for the set value).

Caution To use two capture registers, set the TI000 and TI001 pins.

Remarks 1. For the setting of the TI000 (or TI001) pin, see **8.3 (5) Port mode register 5 (PM5)**.

2. For how to enable the INTTM00 (or INTTM01) interrupt, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

(1) Pulse width measurement with free-running counter and one capture register

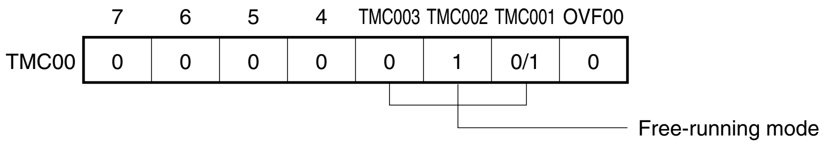
When 16-bit timer counter 00 (TM00) is operated in free-running mode, and the edge specified by prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 01 (CR01) and an external interrupt request signal (INTTM01) is set.

Specify both the rising and falling edges of the TI000 pin by using bits 4 and 5 (ES000 and ES001) of PRM00.

Sampling is performed using the count clock selected by PRM00, and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 8-17. Control Register Settings for Pulse Width Measurement with Free-Running Counter and One Capture Register (When TI000 and CR01 Are Used)

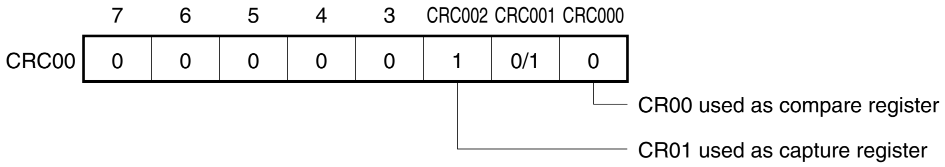
(a) 16-bit timer mode control register 00 (TMC00)



	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

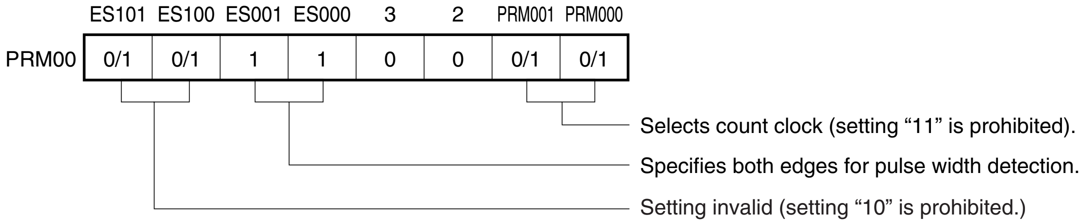


	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	0/1	0

CR00 used as compare register

CR01 used as capture register

(c) Prescaler mode register 00 (PRM00)



	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	1	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).

Specifies both edges for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-18. Configuration Diagram for Pulse Width Measurement with Free-Running Counter

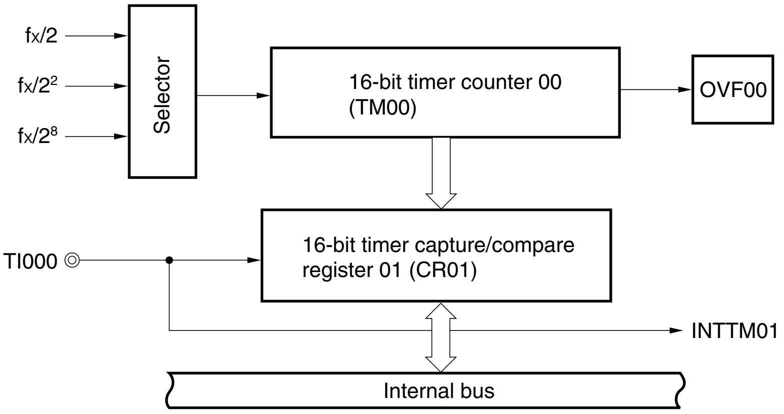
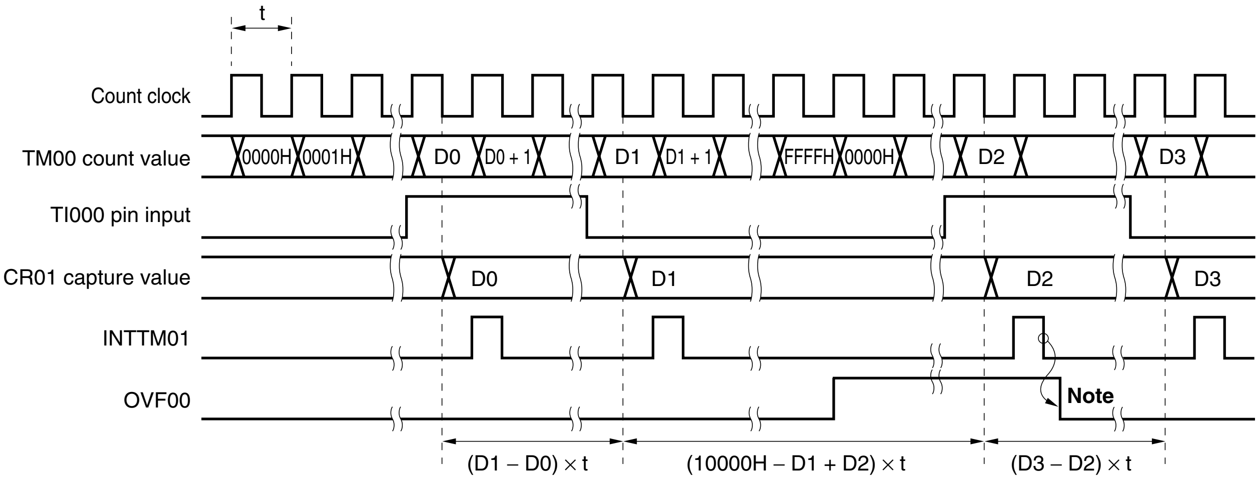

fx/2
fx/2²
fx/2⁸
Selector
16-bit timer counter 00 (TM00)
OVF00
TI000
16-bit timer capture/compare register 01 (CR01)
INTTM01
Internal bus

Figure 8-19. Timing of Pulse Width Measurement Operation with Free-Running Counter and One Capture Register (with Both Edges Specified)



Note Clear OVF00 by software.

(2) Measurement of two pulse widths with free-running counter

When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to simultaneously measure the pulse widths of the two signals input to the TI000 pin and the TI001 pin.

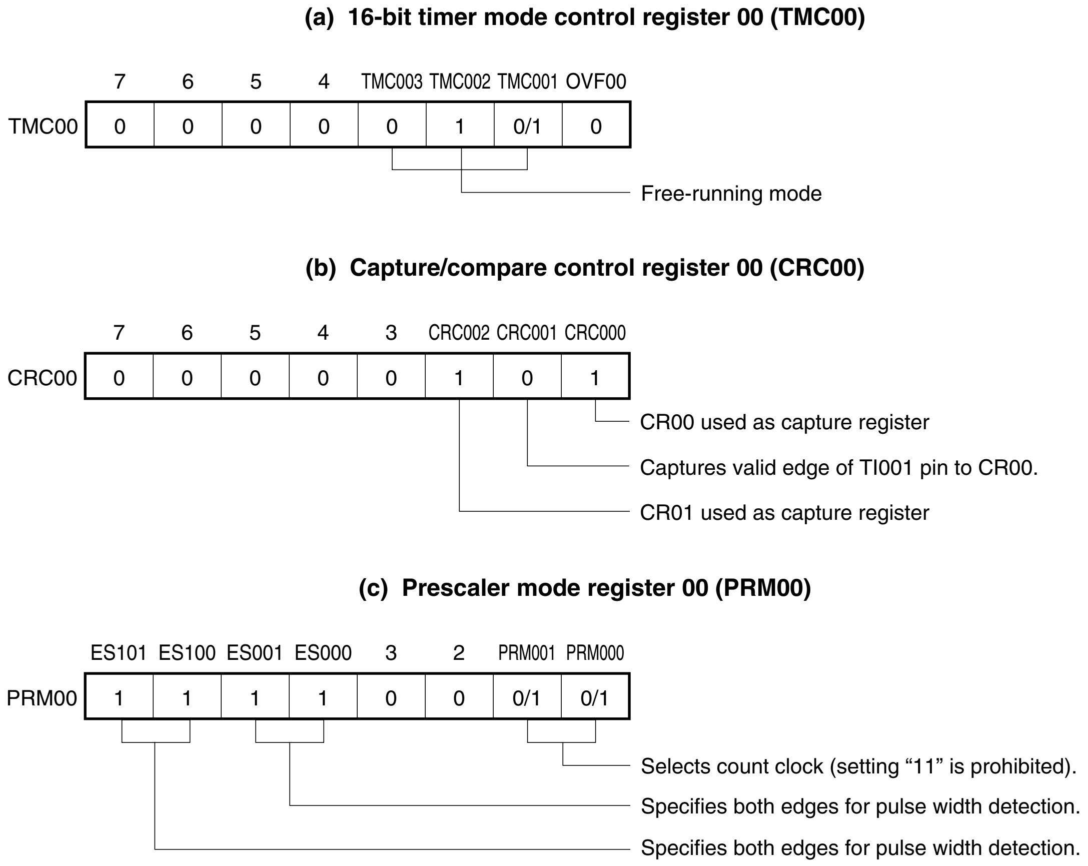
When the edge specified by bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

Also, when the edge specified by bits 6 and 7 (ES100 and ES101) of PRM00 is input to the TI001 pin, the value of TM00 is taken into 16-bit timer capture/compare register 00 (CR00) and an interrupt request signal (INTTM00) is set.

Specify both the rising and falling edges as the edges of the TI000 and TI001 pins, by using bits 4 and 5 (ES000 and ES001) and bits 6 and 7 (ES100 and ES101) of PRM00.

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 or TI001 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 8-20. Control Register Settings for Measurement of Two Pulse Widths with Free-Running Counter



(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	0	1

CR00 used as capture register

Captures valid edge of TI001 pin to CR00.

CR01 used as capture register

(c) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	1	1	1	1	0	0	0/1	0/1

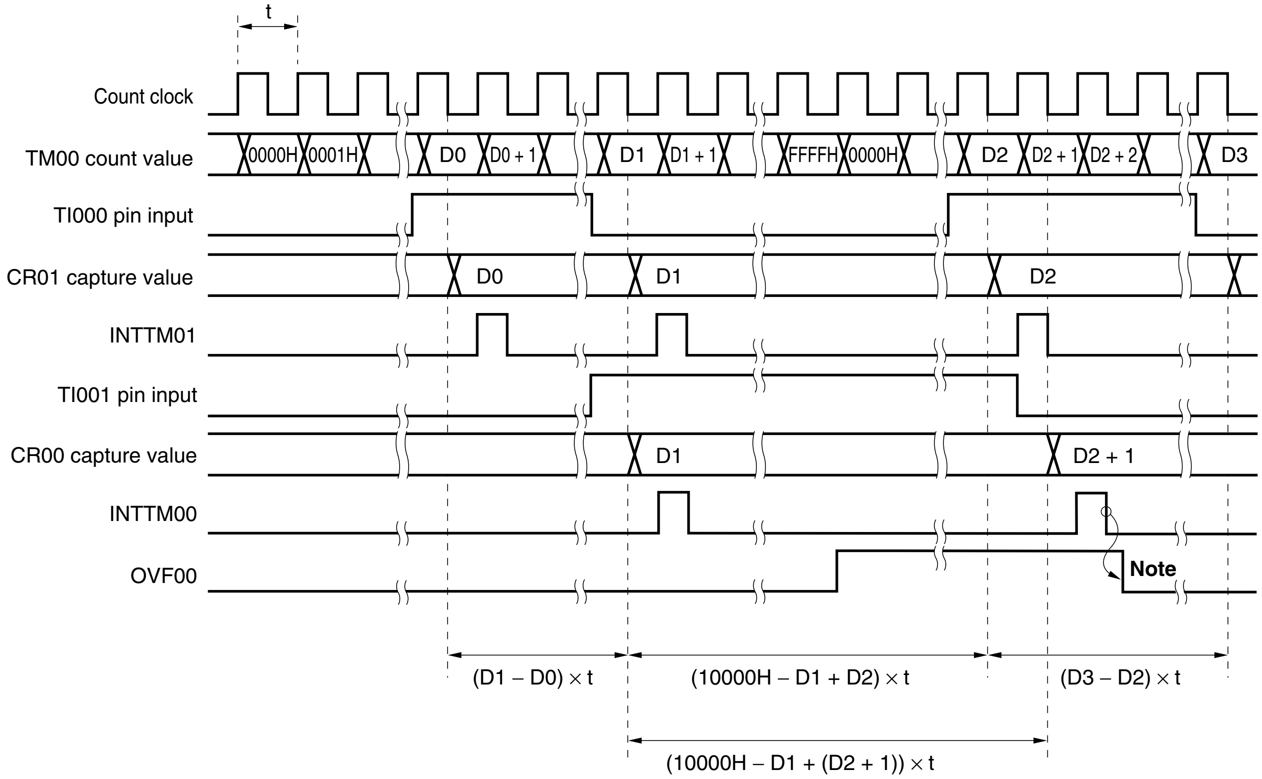
Selects count clock (setting “11” is prohibited).

Specifies both edges for pulse width detection.

Specifies both edges for pulse width detection.

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-21. Timing of Pulse Width Measurement Operation with Free-Running Counter (with Both Edges Specified)



Note Clear OVF00 by software.

(3) Pulse width measurement with free-running counter and two capture registers

When 16-bit timer counter 00 (TM00) is operated in free-running mode, it is possible to measure the pulse width of the signal input to the TI000 pin.

When the rising or falling edge specified by bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00) is input to the TI000 pin, the value of TM00 is taken into 16-bit timer capture/compare register 01 (CR01) and an interrupt request signal (INTTM01) is set.

Also, when the inverse edge to that of the capture operation is input into CR01, the value of TM00 is taken into 16-bit timer capture/compare register 00 (CR00).

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00), and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 8-22. Control Register Settings for Pulse Width Measurement with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0/1	0

Free-running mode

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	1	1

CR00 used as capture register

Captures to CR00 at inverse edge to valid edge of TI000.

CR01 used as capture register

(c) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

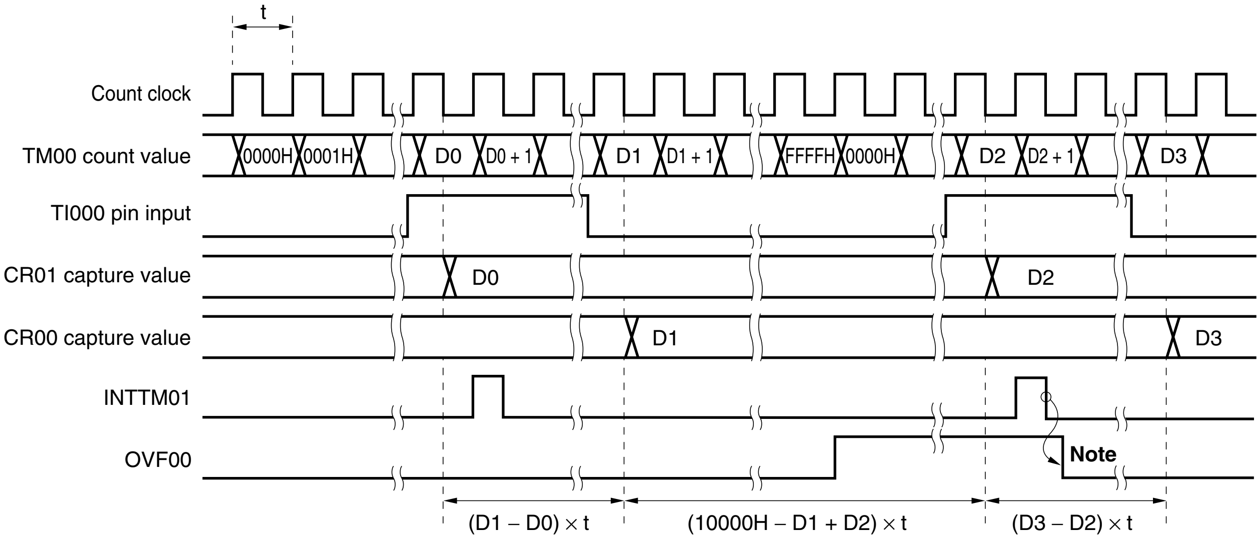
Selects count clock (setting “11” is prohibited).

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with pulse width measurement. See the description of the respective control registers for details.

Figure 8-23. Timing of Pulse Width Measurement Operation with Free-Running Counter and Two Capture Registers (with Rising Edge Specified)



Note Clear OVF00 by software.

(4) Pulse width measurement by means of restart

When input of a valid edge to the TI000 pin is detected, the count value of 16-bit timer counter 00 (TM00) is taken into 16-bit timer capture/compare register 01 (CR01), and then the pulse width of the signal input to the TI000 pin is measured by clearing TM00 and restarting the count operation.

Either of two edges—rising or falling—can be selected using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

Sampling is performed using the count clock cycle selected by prescaler mode register 00 (PRM00) and a capture operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 8-24. Control Register Settings for Pulse Width Measurement by Means of Restart (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	0	0/1	0

Clears and starts at valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	1	1	1

CR00 used as capture register

Captures to CR00 at inverse edge to valid edge of TI000.

CR01 used as capture register

(c) Prescaler mode register 00 (PRM00)

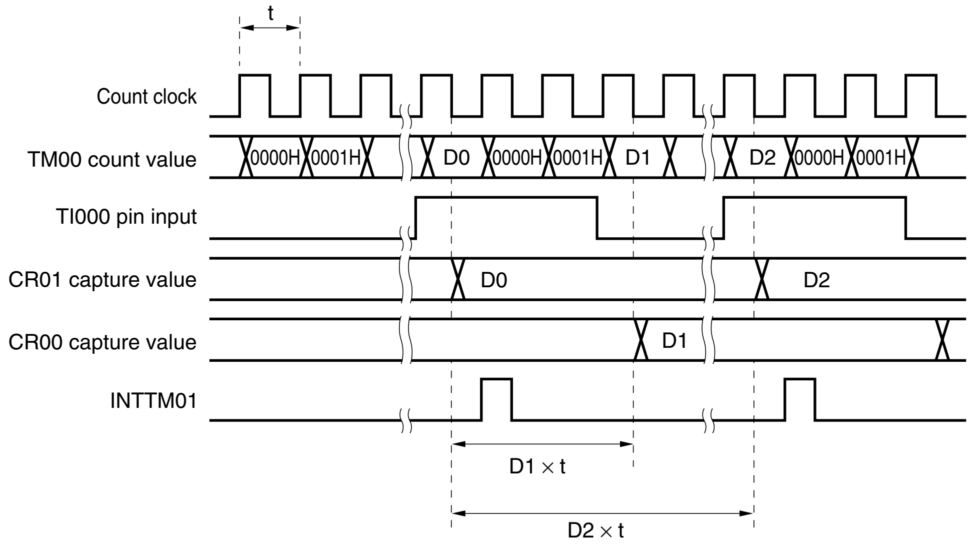
	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

Selects count clock (setting “11” is prohibited).

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Figure 8-25. Timing of Pulse Width Measurement Operation by Means of Restart (with Rising Edge Specified)



8.4.4 External event counter operation

Setting

The basic operation setting procedure is as follows.

<1> Set the CRC00 register (see **Figure 8-26** for the set value).
<2> Set the count clock by using the PRM00 register.
<3> Set any value to the CR00 register (0000H cannot be set).
<4> Set the TMC00 register to start the operation (see **Figure 8-26** for the set value).

Remarks 1. For the setting of the TI000 pin, see **8.3 (5) Port mode register 5 (PM5)**.
2. For how to enable the INTTM00 interrupt, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

The external event counter counts the number of external clock pulses input to the TI000 pin using 16-bit timer counter 00 (TM00).

TM00 is incremented each time the valid edge specified by prescaler mode register 00 (PRM00) is input.

When the TM00 count value matches the 16-bit timer capture/compare register 00 (CR00) value, TM00 is cleared to 0 and the interrupt request signal (INTTM00) is generated.

Input a value other than 0000H to CR00 (a count operation with 1-bit pulse cannot be carried out).

Any of three edges—rising, falling, or both edges—can be selected using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

Sampling is performed using the internal clock (fx) and an operation is only performed when a valid level of the TI000 pin is detected twice, thus eliminating noise with a short pulse width.

Figure 8-26. Control Register Settings in External Event Counter Mode (with Rising Edge Specified)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0/1	0

Clears and starts on match between TM00 and CR00.

(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0/1	0/1	0

CR00 used as compare register

(c) Prescaler mode register 00 (PRM00)

	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	1	1

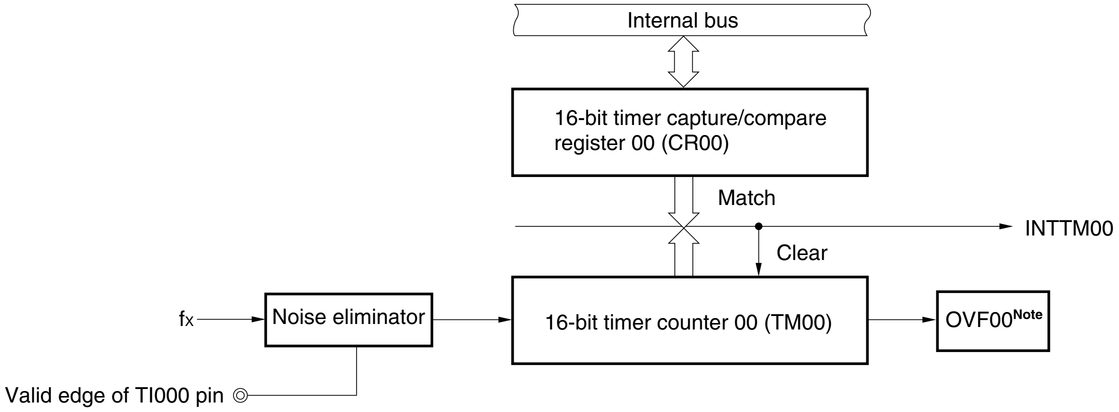
Selects external clock.

Specifies rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

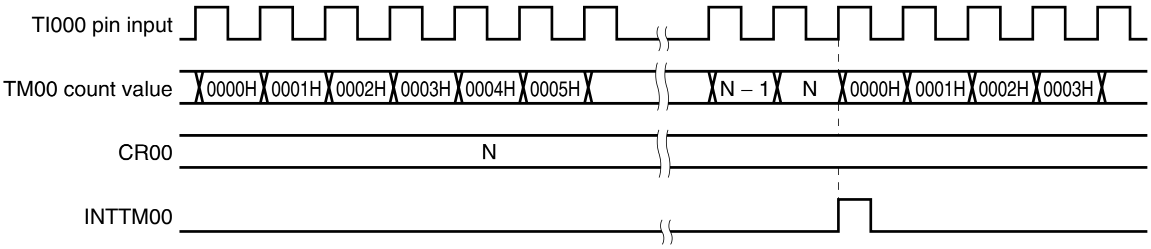
Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with the external event counter. See the description of the respective control registers for details.

Figure 8-27. Configuration Diagram of External Event Counter



Note OVF00 is set to 1 only when CR00 is set to FFFFH.

Figure 8-28. External Event Counter Operation Timing (with Rising Edge Specified)



Caution When reading the external event counter count value, TM00 should be read.

8.4.5 Square-wave output operation

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM00 register.
<2> Set the CRC00 register (see **Figure 8-29** for the set value).
<3> Set the TOC00 register (see **Figure 8-29** for the set value).
<4> Set any value to the CR00 register (0000H cannot be set).
<5> Set the TMC00 register to start the operation (see **Figure 8-29** for the set value).

Caution CR00 cannot be rewritten during TM00 operation.

Remarks 1. For the setting of the TO00 pin, see **8.3 (5) Port mode register 5 (PM5)**.
2. For how to enable the INTTM00 interrupt, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

A square wave with any selected frequency can be output at intervals determined by the count value preset to 16-bit timer capture/compare register 00 (CR00).

The TO00 pin output status is reversed at intervals determined by the count value preset to CR00 + 1 by setting bit 0 (TOE00) and bit 1 (TOC001) of 16-bit timer output control register 00 (TOC00) to 1. This enables a square wave with any selected frequency to be output.

Figure 8-29. Control Register Settings in Square-Wave Output Mode (1/2)

(a) 16-bit timer mode control register 00 (TMC00)

	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	1	0	0

Clears and starts on match between TM00 and CR00.

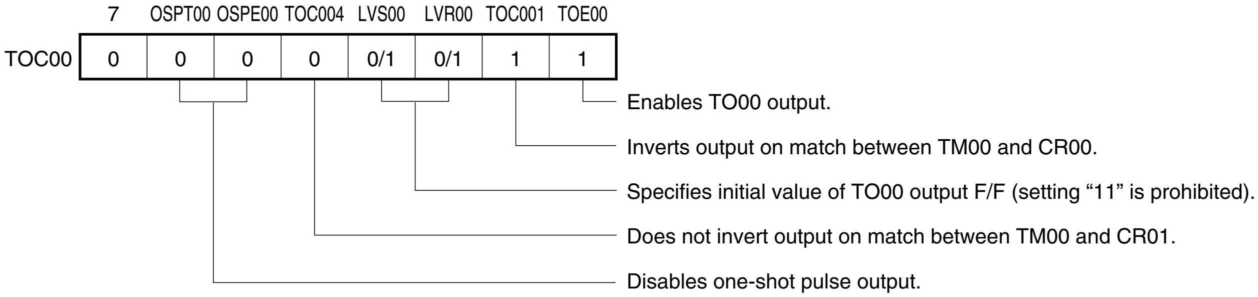
(b) Capture/compare control register 00 (CRC00)

	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0/1	0/1	0

CR00 used as compare register

Figure 8-29. Control Register Settings in Square-Wave Output Mode (2/2)

(c) 16-bit timer output control register 00 (TOC00)



	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	0	0	0/1	0/1	1	1

Enables TO00 output.

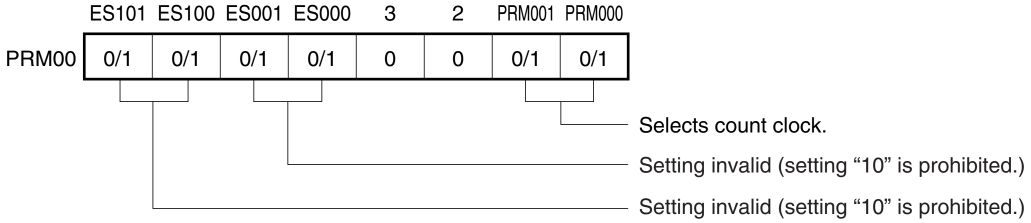
Inverts output on match between TM00 and CR00.

Specifies initial value of TO00 output F/F (setting “11” is prohibited).

Does not invert output on match between TM00 and CR01.

Disables one-shot pulse output.

(d) Prescaler mode register 00 (PRM00)



	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0/1	0/1	0	0	0/1	0/1

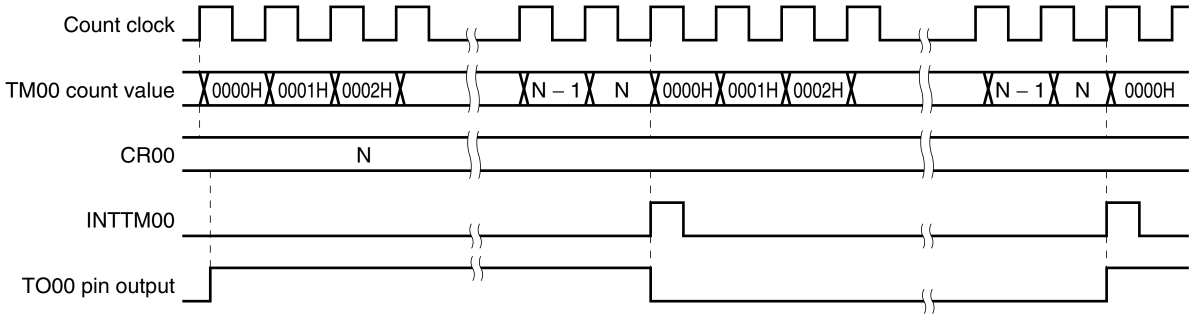
Selects count clock.

Setting invalid (setting “10” is prohibited.)

Setting invalid (setting “10” is prohibited.)

Remark 0/1: Setting 0 or 1 allows another function to be used simultaneously with square-wave output. See the description of the respective control registers for details.

Figure 8-30. Square-Wave Output Operation Timing



8.4.6 One-shot pulse output operation

16-bit timer/event counter 00 can output a one-shot pulse in synchronization with a software trigger or an external trigger (TI000 pin input).

Setting

The basic operation setting procedure is as follows.

<1> Set the count clock by using the PRM00 register.
<2> Set the CRC00 register (see **Figures 8-31** and **8-33** for the set value).
<3> Set the TOC00 register (see **Figures 8-31** and **8-33** for the set value).
<4> Set any value to the CR00 and CR01 registers (0000H cannot be set).
<5> Set the TMC00 register to start the operation (see **Figures 8-31** and **8-33** for the set value).

Remarks 1. For the setting of the TO00 pin, see **8.3 (5) Port mode register 5 (PM5)**.

2. For how to enable the INTTM00 (if necessary, INTTM01) interrupt, see **CHAPTER 19 INTERRUPT FUNCTIONS**.

(1) One-shot pulse output with software trigger

A one-shot pulse can be output from the TO00 pin by setting 16-bit timer mode control register 00 (TMC00), capture/compare control register 00 (CRC00), and 16-bit timer output control register 00 (TOC00) as shown in Figure 8-31, and by setting bit 6 (OSPT00) of the TOC00 register to 1 by software.

By setting the OSPT00 bit to 1, 16-bit timer/event counter 00 is cleared and started, and its output becomes active at the count value (N) set in advance to 16-bit timer capture/compare register 01 (CR01). After that, the output becomes inactive at the count value (M) set in advance to 16-bit timer capture/compare register 00 (CR00)[Note].

Even after the one-shot pulse has been output, the TM00 register continues its operation. To stop the TM00 register, the TMC003 and TMC002 bits of the TMC00 register must be cleared to 00.

Note The case where N < M is described here. When N > M, the output becomes active with the CR00 register and inactive with the CR01 register. Do not set N to M.

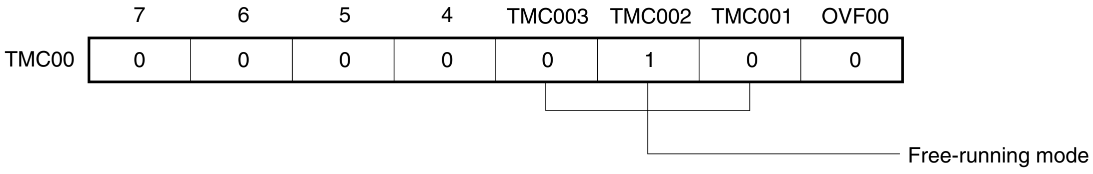
Cautions 1. Do not set the OSPT00 bit while the one-shot pulse is being output. To output the one-shot pulse again, wait until the current one-shot pulse output is completed.

2. When using the one-shot pulse output of 16-bit timer/event counter 00 with a software trigger, do not change the level of the TI000 pin or its alternate-function port pin.

Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate-function port pin, resulting in the output of a pulse at an undesired timing.

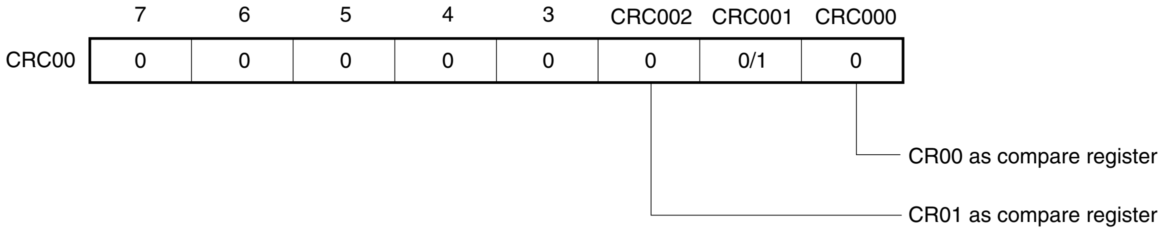
Figure 8-31. Control Register Settings for One-Shot Pulse Output with Software Trigger

(a) 16-bit timer mode control register 00 (TMC00)



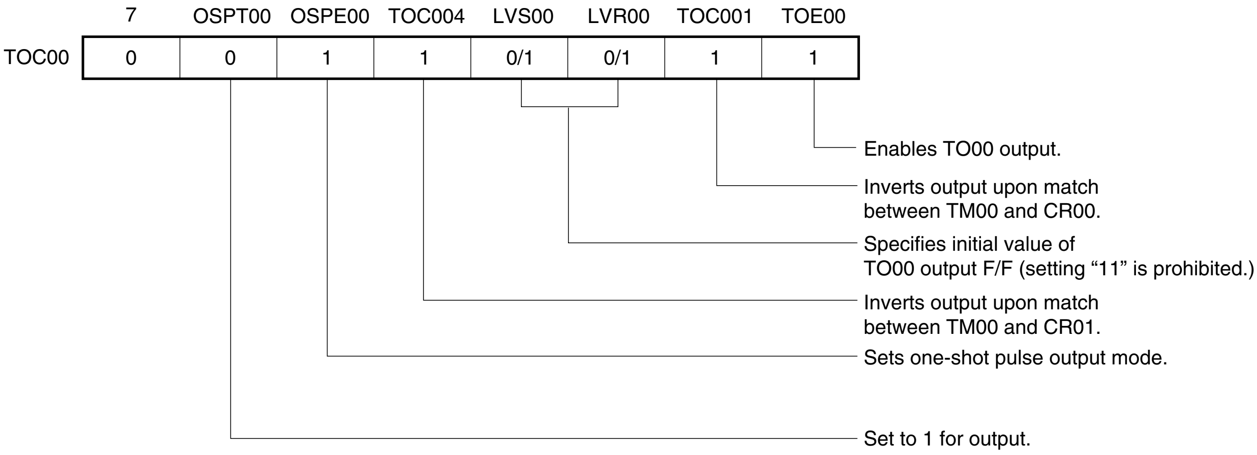
	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	0	1	0	0

(b) Capture/compare control register 00 (CRC00)



	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0	0/1	0

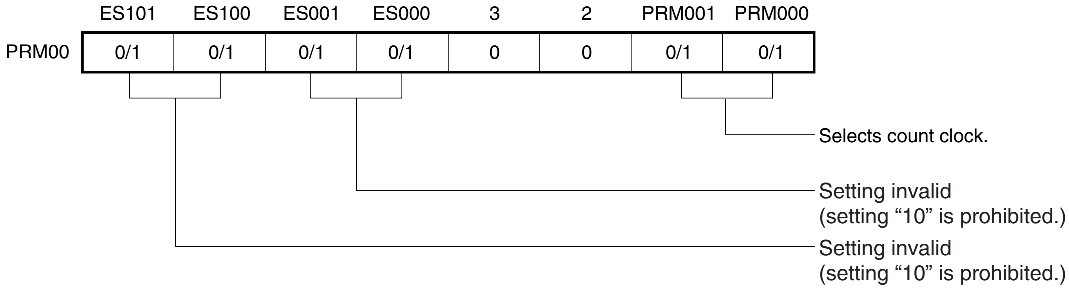
(c) 16-bit timer output control register 00 (TOC00)



	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	1	1	0/1	0/1	1	1

Enables TO00 output.
Inverts output upon match between TM00 and CR00.
Specifies initial value of TO00 output F/F (setting “11” is prohibited.)
Inverts output upon match between TM00 and CR01.
Sets one-shot pulse output mode.
Set to 1 for output.

(d) Prescaler mode register 00 (PRM00)

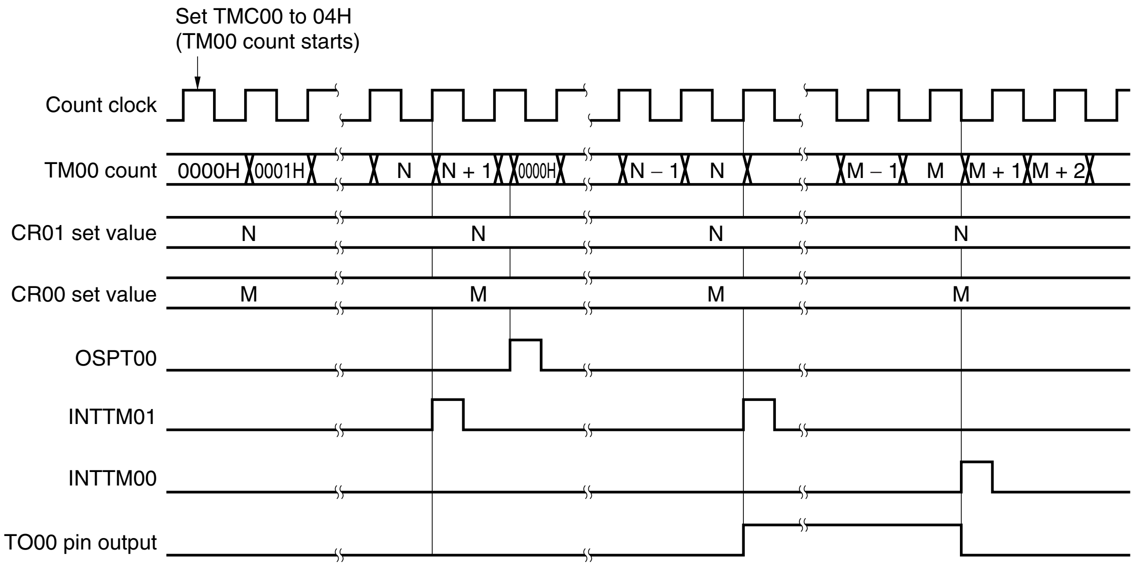


	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0/1	0/1	0	0	0/1	0/1

Selects count clock.
Setting invalid (setting “10” is prohibited.)
Setting invalid (setting “10” is prohibited.)

Caution Do not set 0000H to the CR00 and CR01 registers.

Figure 8-32. Timing of One-Shot Pulse Output Operation with Software Trigger



Caution **16-bit timer counter 00 starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC003 and TMC002 bits.**

Remark N < M

(2) One-shot pulse output with external trigger

A one-shot pulse can be output from the TO00 pin by setting 16-bit timer mode control register 00 (TMC00), capture/compare control register 00 (CRC00), and 16-bit timer output control register 00 (TOC00) as shown in Figure 8-33, and by using the valid edge of the TI000 pin as an external trigger.

The valid edge of the TI000 pin is specified by bits 4 and 5 (ES000, ES001) of prescaler mode register 00 (PRM00). The rising, falling, or both the rising and falling edges can be specified.

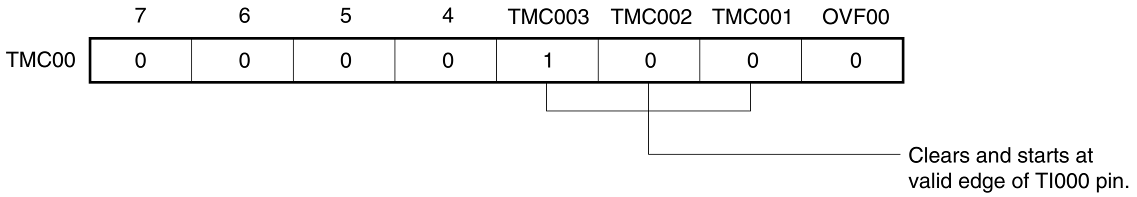
When the valid edge of the TI000 pin is detected, the 16-bit timer/event counter is cleared and started, and the output becomes active at the count value set in advance to 16-bit timer capture/compare register 01 (CR01). After that, the output becomes inactive at the count value set in advance to 16-bit timer capture/compare register 00 (CR00)[Note].

Note The case where N < M is described here. When N > M, the output becomes active with the CR00 register and inactive with the CR01 register. Do not set N to M.

Caution **Even if the external trigger is generated again while the one-shot pulse is output, it is ignored.**

Figure 8-33. Control Register Settings for One-Shot Pulse Output with External Trigger (with Rising Edge Specified)

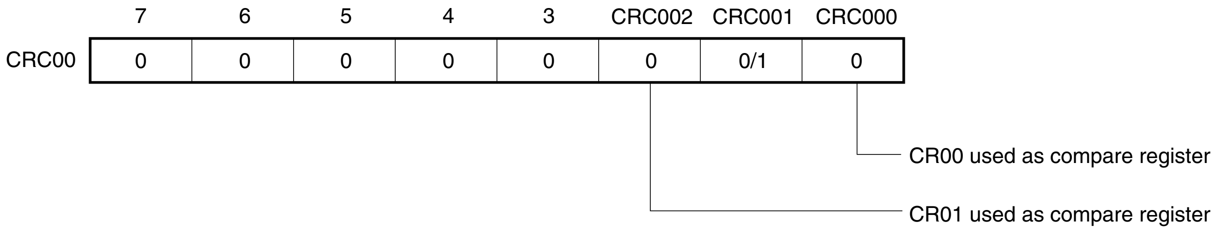
(a) 16-bit timer mode control register 00 (TMC00)



	7	6	5	4	TMC003	TMC002	TMC001	OVF00
TMC00	0	0	0	0	1	0	0	0

Clears and starts at valid edge of TI000 pin.

(b) Capture/compare control register 00 (CRC00)

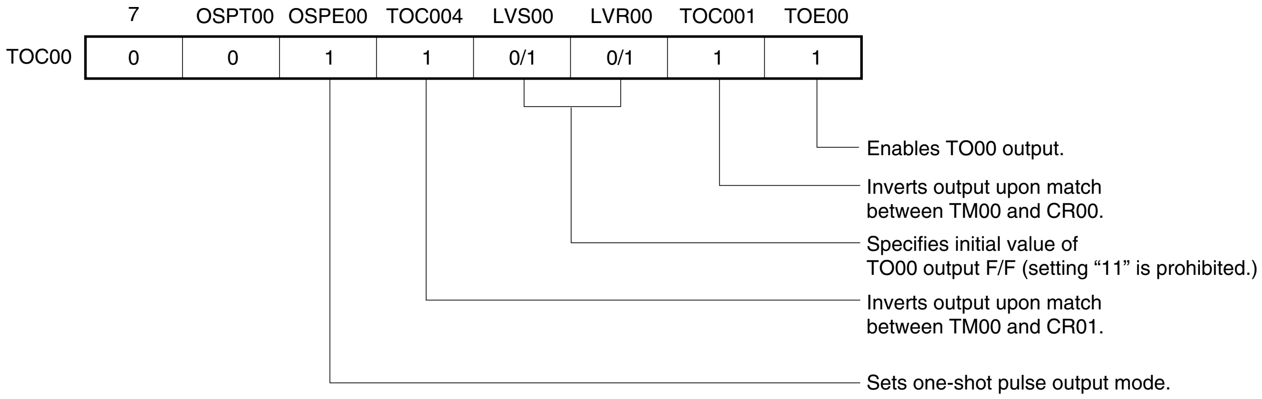


	7	6	5	4	3	CRC002	CRC001	CRC000
CRC00	0	0	0	0	0	0	0/1	0

CR00 used as compare register

CR01 used as compare register

(c) 16-bit timer output control register 00 (TOC00)



	7	OSPT00	OSPE00	TOC004	LVS00	LVR00	TOC001	TOE00
TOC00	0	0	1	1	0/1	0/1	1	1

Enables TO00 output.

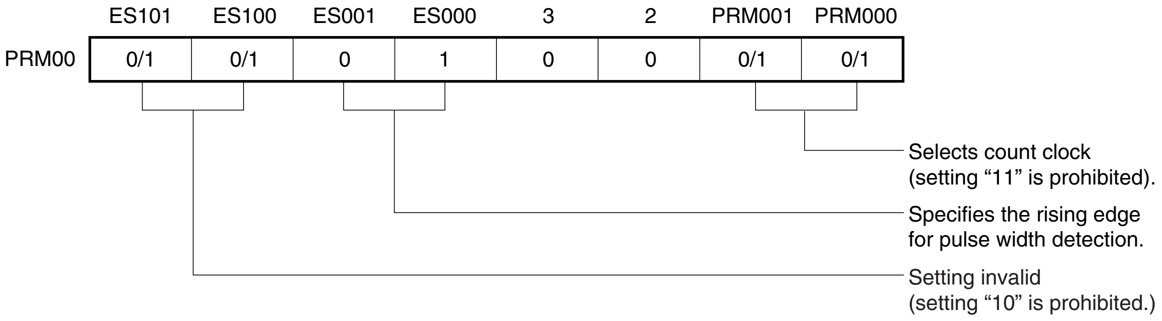
Inverts output upon match between TM00 and CR00.

Specifies initial value of TO00 output F/F (setting “11” is prohibited.)

Inverts output upon match between TM00 and CR01.

Sets one-shot pulse output mode.

(d) Prescaler mode register 00 (PRM00)



	ES101	ES100	ES001	ES000	3	2	PRM001	PRM000
PRM00	0/1	0/1	0	1	0	0	0/1	0/1

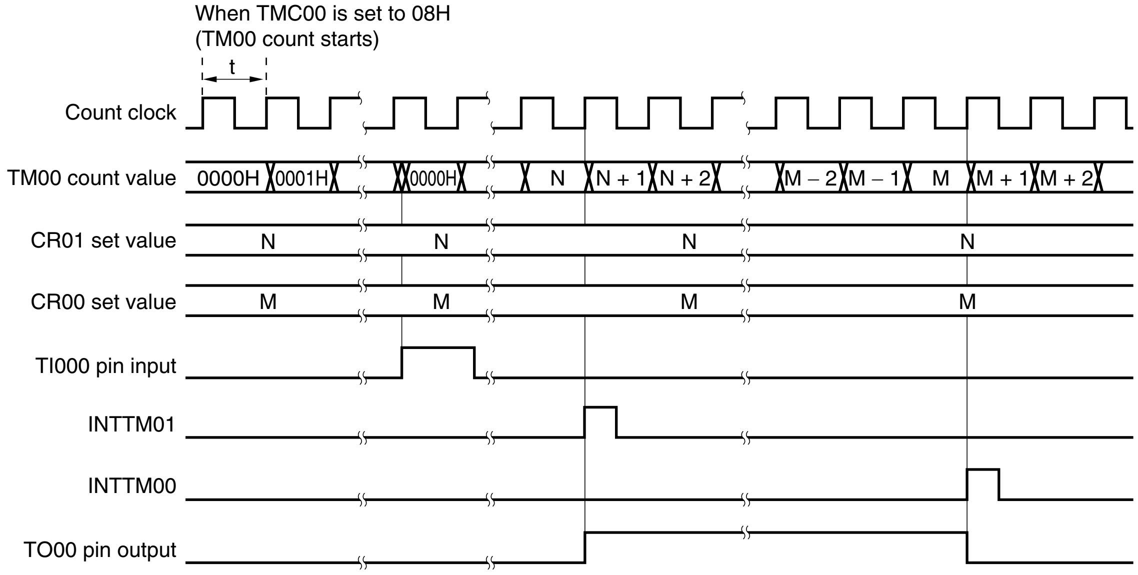
Selects count clock (setting “11” is prohibited).

Specifies the rising edge for pulse width detection.

Setting invalid (setting “10” is prohibited.)

Caution Do not set 0000H to the CR00 and CR01 registers.

Figure 8-34. Timing of One-Shot Pulse Output Operation with External Trigger (with Rising Edge Specified)



Caution 16-bit timer counter 00 starts operating as soon as a value other than 00 (operation stop mode) is set to the TMC002 and TMC003 bits.

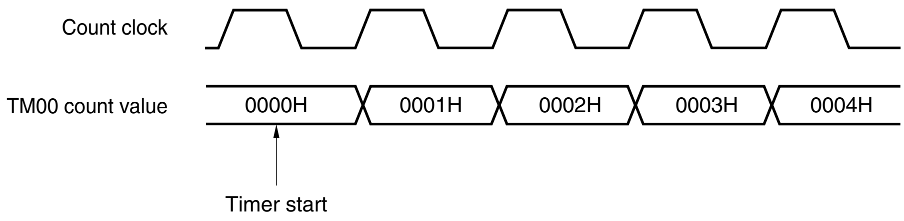
Remark N < M

8.5 Cautions for 16-Bit Timer/Event Counter 00

(1) Timer start errors

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 16-bit timer counter 00 (TM00) is started asynchronously to the count clock.

Figure 8-35. Start Timing of 16-Bit Timer Counter 00 (TM00)



(2) 16-bit timer capture/compare register 00 setting

In the mode in which clear & start occurs on match between TM00 and CR00, set 16-bit timer capture/compare register 00 (CR00) to other than 0000H. This means a 1-pulse count operation cannot be performed when 16-bit timer/event counter 00 is used as an external event counter.

(3) Capture register data retention timing

The values of 16-bit timer capture/compare registers 00 and 01 (CR00 and CR01) are not guaranteed after 16-bit timer/event counter 00 has been stopped.

(4) Valid edge setting

Set the valid edge of the TI000 pin after clearing bits 2 and 3 (TMC002 and TMC003) of 16-bit timer mode control register 00 (TMC00) to 0, 0, respectively, and then stopping timer operation. The valid edge is set using bits 4 and 5 (ES000 and ES001) of prescaler mode register 00 (PRM00).

(5) Re-triggering one-shot pulse

(a) One-shot pulse output by software

When a one-shot pulse is output, do not set the OSPT00 bit to 1. Do not output the one-shot pulse again until INTTM00, which occurs upon a match with the CR00 register, or INTTM01, which occurs upon a match with the CR01 register, occurs.

(b) One-shot pulse output with external trigger

If the external trigger occurs again while a one-shot pulse is output, it is ignored.

(c) One-shot pulse output function

When using the one-shot pulse output of 16-bit timer/event counter 00 with a software trigger, do not change the level of the TI000 pin or its alternate function port pin.

Because the external trigger is valid even in this case, the timer is cleared and started even at the level of the TI000 pin or its alternate function port pin, resulting in the output of a pulse at an undesired timing.

(6) Operation of OVF00 flag

<1> The OVF00 flag is also set to 1 in the following case.

When any of the following modes is selected: the mode in which clear & start occurs on a match between TM00 and CR00, the mode in which clear & start occurs at the TI000 pin valid edge, or the free-running mode

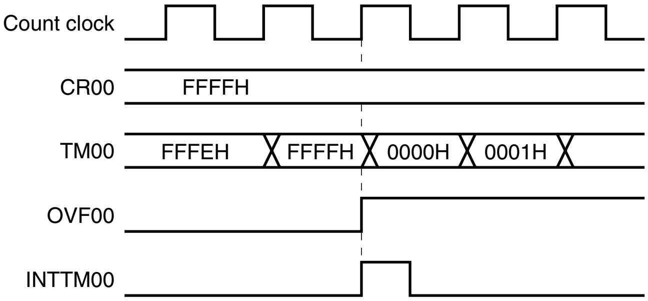
↓

CR00 is set to FFFFH

↓

TM00 is counted up from FFFFH to 0000H.

Figure 8-36. Operation Timing of OVF00 Flag



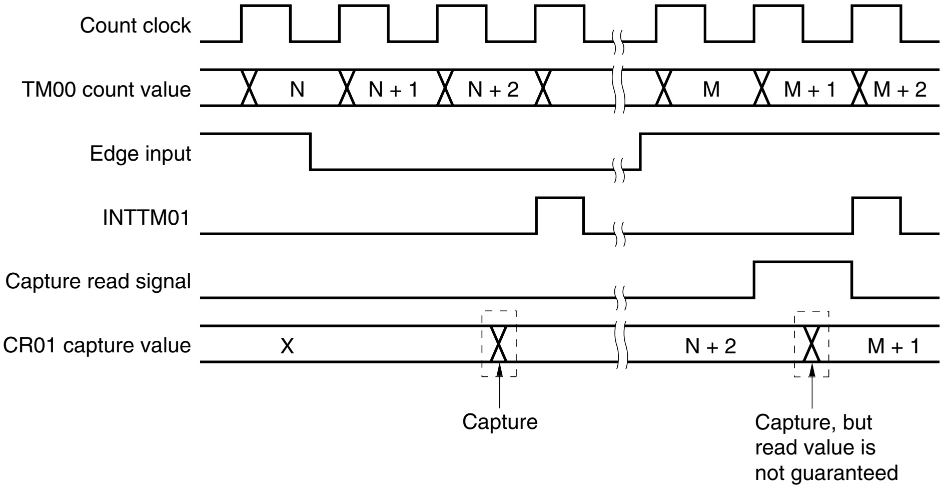
<2> Even if the OVF00 flag is cleared before the next count clock (before TM00 becomes 0001H) after the occurrence of TM00 overflow, the OVF00 flag is re-set newly and clear is disabled.

(7) Conflicting operations

Conflict between the read period of the 16-bit timer capture/compare register (CR00/CR01) and capture trigger input (CR00/CR01 used as capture register)

Capture trigger input has priority. The data read from CR00/CR01 is undefined.

Figure 8-37. Capture Register Data Retention Timing



(8) Timer operation

<1> Even if 16-bit timer counter 00 (TM00) is read, the value is not captured by 16-bit timer capture/compare register 01 (CR01).

<2> Regardless of the CPU's operation mode, when the timer stops, the input signals to the TI000/TI001 pins are not acknowledged.

<3> The one-shot pulse output mode operates correctly only in the free-running mode and the mode in which clear & start occurs at the TI000 valid edge. In the mode in which clear & start occurs on a match between the TM00 register and CR00 register, one-shot pulse output is not possible because an overflow does not occur.

(9) Capture operation

<1> If the TI000 pin valid edge is specified as the count clock, a capture operation by the capture register specified as the trigger for TI000 is not possible.

<2> To ensure the reliability of the capture operation, the capture trigger requires a pulse two cycles longer than the count clock selected by prescaler mode register 00 (PRM00).

<3> The capture operation is performed at the falling edge of the count clock. An interrupt request input (INTTM00/INTTM01), however, is generated at the rise of the next count clock.

(10) Compare operation

A capture operation may not be performed for CR00/CR01 set in compare mode even if a capture trigger has been input.

(11) Edge detection

<1> If the TI000 or TI001 pin is high level immediately after system reset and the rising edge or both the rising and falling edges are specified as the valid edge of the TI000 or TI001 pin to enable the 16-bit timer counter 00 (TM00) operation, a rising edge is detected immediately after the operation is enabled. Be careful therefore when pulling up the TI000 or TI001 pin. However, the rising edge is not detected at restart after the operation has been stopped once.

<2> The sampling clock used to remove noise differs when the TI000 pin valid edge is used as the count clock and when it is used as a capture trigger. In the former case, the count clock is f_X, and in the latter case the count clock is selected by prescaler mode register 00 (PRM00). The capture operation is only performed when a valid level is detected twice by sampling the valid edge, thus eliminating noise with a short pulse width.

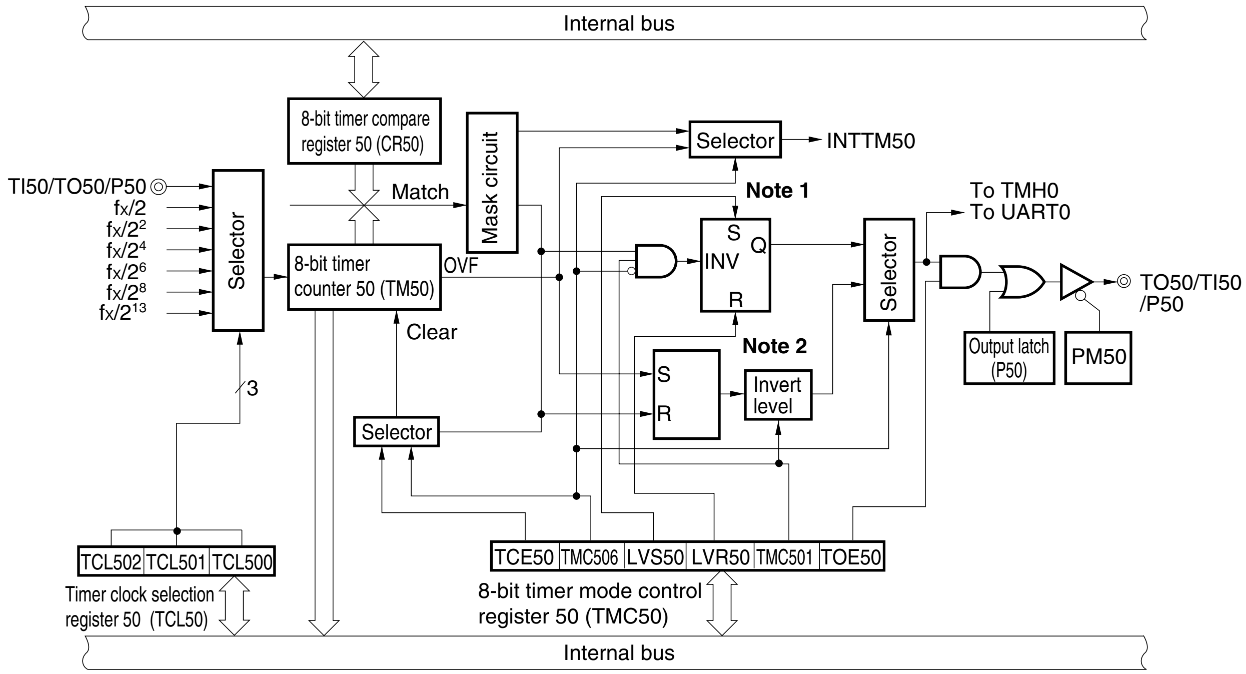
9.1 Functions of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 have the following functions.

- Interval timer
- External event counter
- Square-wave output
- PWM output

Figures 9-1 and 9-2 show the block diagrams of 8-bit timer/event counters 50 and 51.

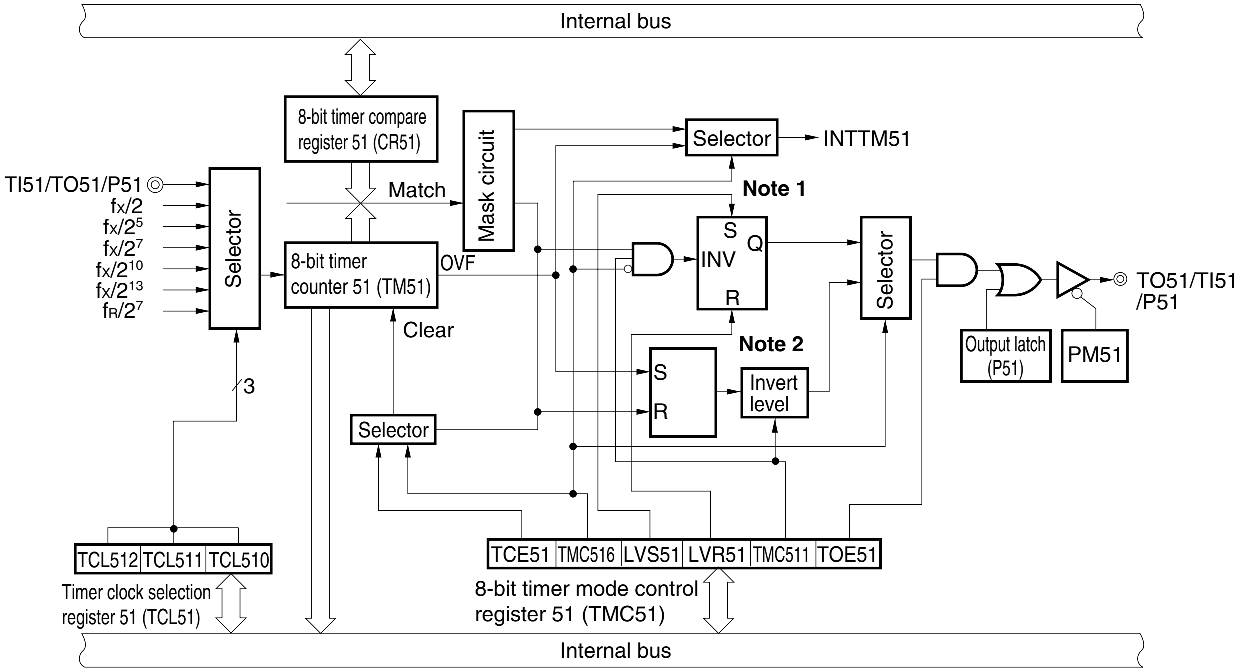
Figure 9-1. Block Diagram of 8-Bit Timer/Event Counter 50



Notes 1. Timer output F/F

2. PWM output F/F

Figure 9-2. Block Diagram of 8-Bit Timer/Event Counter 51



Notes 1. Timer output F/F

2. PWM output F/F

9.2 Configuration of 8-Bit Timer/Event Counters 50 and 51

8-bit timer/event counters 50 and 51 consist of the following hardware.

Table 9-1. Configuration of 8-Bit Timer/Event Counters 50 and 51

Item	Configuration
Timer register	8-bit timer counter 5n (TM5n)
Register	8-bit timer compare register 5n (CR5n)
Timer input	TI5n
Timer output	TO5n
Control registers	Timer clock selection register 5n (TCL5n) 8-bit timer mode control register 5n (TMC5n) Port mode register 5 (PM5) Port register 5 (P5)

(1) 8-bit timer counter 5n (TM5n)

TM5n is an 8-bit register that counts the count pulses and is read-only.

The counter is incremented in synchronization with the rising edge of the count clock.

Figure 9-3. Format of 8-Bit Timer Counter 5n (TM5n)

Address: FF2CH (TM50), FF3CH (TM51) After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
TM5n (n = 0, 1)								

In the following situations, the count value is cleared to 00H.

<1> $\overline{\text{RESET}}$ input

<2> When TCE5n is cleared

<3> When TM5n and CR5n match in the mode in which clear & start occurs upon a match of the TM5n and CR5n.

(2) 8-bit timer compare register 5n (CR5n)

CR5n can be read and written by an 8-bit memory manipulation instruction.

Except in PWM mode, the value set in CR5n is constantly compared with the 8-bit timer counter 5n (TM5n) count value, and an interrupt request (INTTM5n) is generated if they match.

In PWM mode, when the TO5n pin becomes active due to a TM5n overflow and the values of TM5n and CR5n match, the TO5n pin becomes inactive.

The value of CR5n can be set within 00H to FFH.

$\overline{\text{RESET}}$ input clears CR5n to 00H.

Figure 9-4. Format of 8-Bit Timer Compare Register 5n (CR5n)

Address: FF2DH (CR50), FF3DH (CR51) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CR5n (n = 0, 1)								

Cautions 1. In the mode in which clear & start occurs on a match of TM5n and CR5n (TMC5n6 = 0), do not write other values to CR5n during operation.

2. In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

9.3 Registers Controlling 8-Bit Timer/Event Counters 50 and 51

The following four registers are used to control 8-bit timer/event counters 50 and 51.

- Timer clock selection register 5n (TCL5n)
- 8-bit timer mode control register 5n (TMC5n)
- Port mode register 5 (PM5)
- Port register 5 (P5)

(1) Timer clock selection register 5n (TCL5n)

This register sets the count clock of 8-bit timer/event counter 5n and the valid edge of the TI5n pin input.
TCL5n can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears TCL5n to 00H.

Remark n = 0, 1

Figure 9-5. Format of Timer Clock Selection Register 50 (TCL50)

Address: FF2EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL50	0	0	0	0	0	TCL502	TCL501	TCL500

TCL502	TCL501	TCL500	Count clock selection [Note]
0	0	0	TI50 pin falling edge
0	0	1	TI50 pin rising edge
0	1	0	$f_X/2$ (10 MHz)
0	1	1	$f_X/2^{2}$ (5 MHz)
1	0	0	$f_X/2^{4}$ (1.25 MHz)
1	0	1	$f_X/2^{6}$ (312.5 kHz)
1	1	0	$f_X/2^{8}$ (78.125 kHz)
1	1	1	$f_X/2^{13}$ (2.44 kHz)

Note Be sure to set the count clock so that the following condition is satisfied.
- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz

Cautions 1. When the internal oscillation clock is selected as the source clock to the CPU, the clock of the internal oscillator is divided and supplied as the count clock. If the count clock is the internal oscillation clock, the operation of 8-bit timer/event counter 50 is not guaranteed.

2. When rewriting TCL50 to other data, stop the timer operation beforehand.

3. Be sure to clear bits 3 to 7 to 0.

Remarks 1. f_X: X1 input clock oscillation frequency

2. Figures in parentheses apply to operation at f_X = 20 MHz.

Figure 9-6. Format of Timer Clock Selection Register 51 (TCL51)

Address: FF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TCL51	0	0	0	0	0	TCL512	TCL511	TCL510

TCL512	TCL511	TCL510	Count clock selection [Note]
0	0	0	TI51 pin falling edge
0	0	1	TI51 pin rising edge
0	1	0	$f_X/2$ (10 MHz)
0	1	1	$f_X/2^5$ (625 kHz)
1	0	0	$f_X/2^7$ (156.25 kHz)
1	0	1	$f_X/2^{10}$ (19.53 kHz)
1	1	0	$f_X/2^{13}$ (2.44 kHz)
1	1	1	$f_R/2^7$ (1.88 kHz)

Note Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz

Cautions 1. When the internal oscillation clock is selected as the source clock to the CPU, the clock of the internal oscillator is divided and supplied as the count clock. If the count clock is the internal oscillation clock, the operation of 8-bit timer/event counter 51 is not guaranteed.

2. When rewriting TCL51 to other data, stop the timer operation beforehand.

3. Be sure to clear bits 3 to 7 to 0.

Remarks 1. f_X: X1 input clock oscillation frequency

f_R: internal oscillation clock frequency

2. Figures in parentheses apply to operation at f_X = 20 MHz and f_R = 240 kHz (typ.).

(2) 8-bit timer mode control register 5n (TMC5n)

TMC5n is a register that performs the following five types of settings.

<1> 8-bit timer counter 5n (TM5n) count operation control
<2> 8-bit timer counter 5n (TM5n) operating mode selection
<3> Timer output F/F (flip-flop) status setting
<4> Active level selection in timer F/F control or PWM (free-running) mode
<5> Timer output control

TMC5n can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Remark n = 0, 1

Figure 9-7. Format of 8-Bit Timer Mode Control Register 50 (TMC50)

Address: FF2FH After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC50	TCE50	TMC506	0	0	LVS50	LVR50	TMC501	TOE50

TCE50	TM50 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC506	TM50 operating mode selection
0	Mode in which clear & start occurs on a match between TM50 and CR50
1	PWM (free-running) mode

LVS50	LVR50	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC501	In other modes (TMC506 = 0)	In PWM mode (TMC506 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE50	Timer output control
0	Output disabled (TM50 output is low level)
1	Output enabled

(Refer to the next page for **Caution** and **Remark**.)

Figure 9-8. Format of 8-Bit Timer Mode Control Register 51 (TMC51)

Address: FF3FH After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	1	<0>
TMC51	TCE51	TMC516	0	0	LVS51	LVR51	TMC511	TOE51

TCE51	TM51 count operation control
0	After clearing to 0, count operation disabled (counter stopped)
1	Count operation start

TMC516	TM51 operating mode selection
0	Mode in which clear & start occurs on a match between TM51 and CR51
1	PWM (free-running) mode

LVS51	LVR51	Timer output F/F status setting
0	0	No change
0	1	Timer output F/F reset (0)
1	0	Timer output F/F set (1)
1	1	Setting prohibited

TMC511	In other modes (TMC516 = 0)	In PWM mode (TMC516 = 1)
	Timer F/F control	Active level selection
0	Inversion operation disabled	Active-high
1	Inversion operation enabled	Active-low

TOE51	Timer output control
0	Output disabled (TM51 output is low level)
1	Output enabled

Cautions 1. The settings of LVS5n and LVR5n are valid in other than PWM mode.

2. Do not rewrite following bits simultaneously.

- **TMC5n1 and TOE5n**
- **TMC5n6 and TOE5n**
- **TMC5n1 and TMC5n6**
- **TMC5n6 and LVS5n, LVR5n**
- **TOE5n and LVS5n, LVR5n**

3. Stop operation before rewriting TMC5n6.

Remarks 1. In PWM mode, PWM output is made inactive by clearing TCE5n to 0.

2. If LVS5n and LVR5n are read, the value is 0.

3. The values of the TMC5n6, LVS5n, LVR5n, TMC5n1, and TOE5n bits are reflected at the TO5n pin regardless of the value of TCE5n.

4. n = 0, 1

(3) Port mode register 5 (PM5)

This register sets port 5 input/output in 1-bit units.

When using the P50/TO50/TI50 and P51/TO51/TI51 pins for timer output, clear PM50 and PM51 and the output latches of P50 and P51 to 0.

When using the P50/TO50/TI50 and P51/TO51/TI51 pins for timer input, set PM50 and PM51 to 1. The output latches of P50 and P51 at this time may be 0 or 1.

PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to FFH.

Figure 9-9. Format of Port Mode Register 5 (PM5)

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	P5n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Operations of 8-Bit Timer/Event Counters 50 and 51

9.4.1 Operation as interval timer

8-bit timer/event counter 5n operates as an interval timer that generates interrupt requests repeatedly at intervals of the count value preset to 8-bit timer compare register 5n (CR5n).

When the count value of 8-bit timer counter 5n (TM5n) matches the value set to CR5n, counting continues with the TM5n value cleared to 0 and an interrupt request signal (INTTM5n) is generated.

The count clock of TM5n can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

Setting

<1> Set the registers.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.
  (TMC5n = 0000×××0B × = Don’t care)

<2> After TCE5n = 1 is set, the count operation starts.

<3> If the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

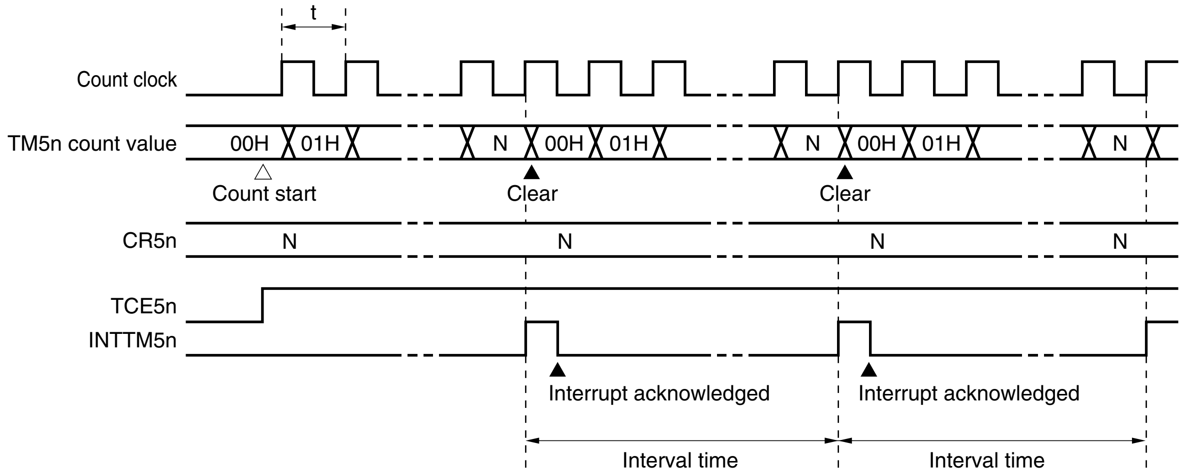
<4> INTTM5n is generated repeatedly at the same interval.
Clear TCE5n to 0 to stop the count operation.

Caution Do not write other values to CR5n during operation.

Remark n = 0, 1

Figure 9-10. Interval Timer Operation Timing (1/2)

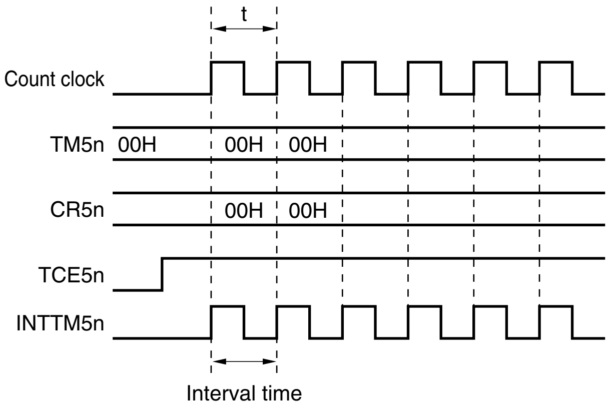
(a) Basic operation



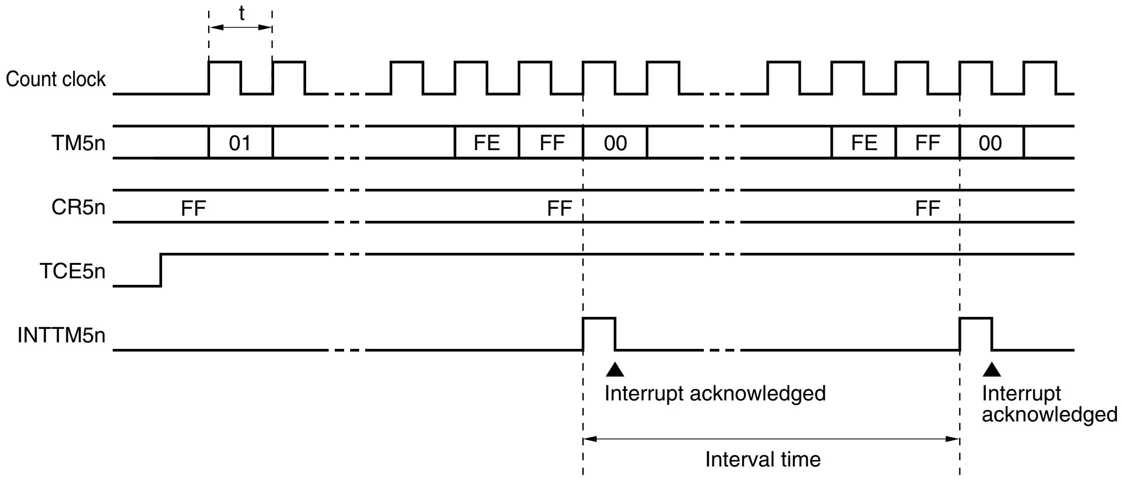
Remark Interval time = (N + 1) × t
N = 00H to FFH
n = 0, 1

Figure 9-10. Interval Timer Operation Timing (2/2)

(b) When CR5n = 00H



(c) When CR5n = FFH



Remark n = 0, 1

9.4.2 Operation as external event counter

The external event counter counts the number of external clock pulses to be input to the TI5n pin by 8-bit timer counter 5n (TM5n).

TM5n is incremented each time the valid edge specified by timer clock selection register 5n (TCL5n) is input. Either the rising or falling edge can be selected.

When the TM5n count value matches the value of 8-bit timer compare register 5n (CR5n), TM5n is cleared to 0 and an interrupt request signal (INTTM5n) is generated.

Whenever the TM5n value matches the value of CR5n, INTTM5n is generated.

Setting

<1> Set each register.

- Set the port mode register (PM50 or PM51)[Note] to 1.
- TCL5n: Select TI5n pin input edge.
  TI5n pin falling edge → TCL5n = 00H
  TI5n pin rising edge → TCL5n = 01H
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on match of TM5n and CR5n, disable the timer F/F inversion operation, disable timer output.
  (TMC5n = 0000××00B × = Don't care)

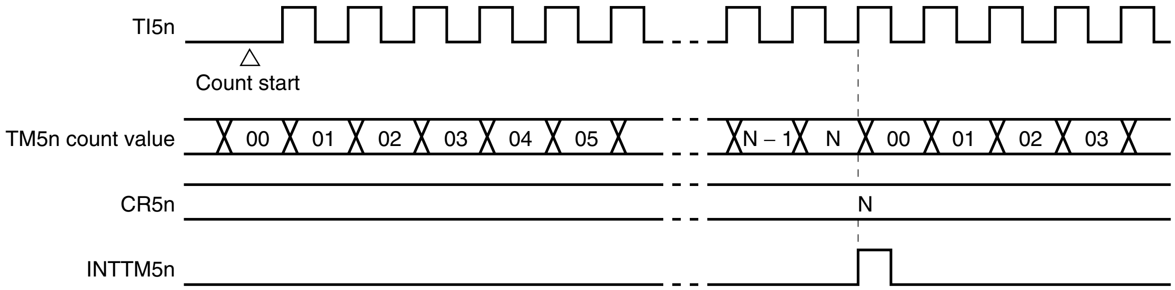
<2> When TCE5n = 1 is set, the number of pulses input from the TI5n pin is counted.

<3> When the values of TM5n and CR5n match, INTTM5n is generated (TM5n is cleared to 00H).

<4> After these settings, INTTM5n is generated each time the values of TM5n and CR5n match.

Note 8-bit timer/event counter 50: PM50
8-bit timer/event counter 51: PM51

Figure 9-11. External Event Counter Operation Timing (with Rising Edge Specified)



Remark N = 00H to FFH
n = 0, 1

9.4.3 Square-wave output operation

A square wave with any selected frequency is output at intervals determined by the value preset to 8-bit timer compare register 5n (CR5n).

The TO5n pin output status is inverted at intervals determined by the count value preset to CR5n by setting bit 0 (TOE5n) of 8-bit timer mode control register 5n (TMC5n) to 1. This enables a square wave with any selected frequency to be output (duty = 50%).

Setting

<1> Set each register.

- Clear the port output latch (P50 or P51)[Note] and port mode register (PM50 or PM51)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select the mode in which clear & start occurs on a match of TM5n and CR5n.

LVS5n	LVR5n	Timer Output F/F Status Setting
1	0	High-level output
0	1	Low-level output

Timer output F/F inversion enabled
Timer output enabled
(TMC5n = 00001011B or 00000111B)

<2> After TCE5n = 1 is set, the count operation starts.

<3> The timer output F/F is inverted by a match of TM5n and CR5n. After INTTM5n is generated, TM5n is cleared to 00H.

<4> After these settings, the timer output F/F is inverted at the same interval and a square wave is output from TO5n.

The frequency is as follows.

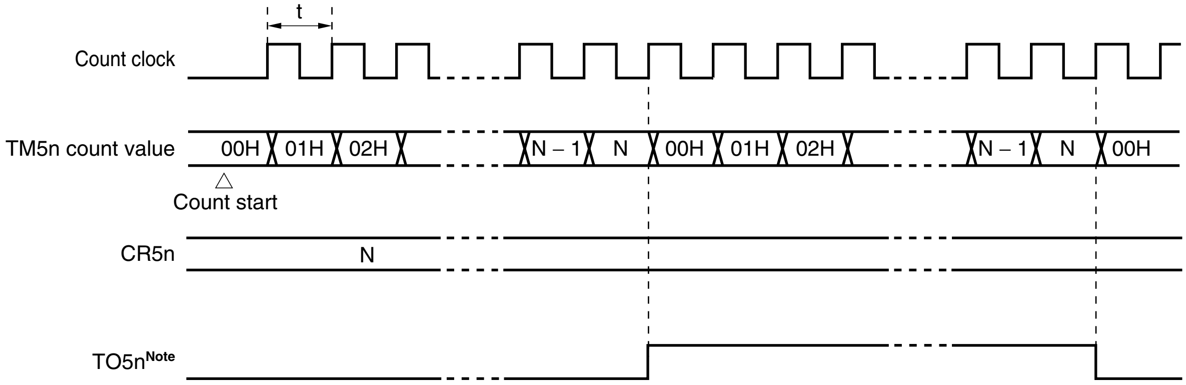
Frequency = 1/2t (N + 1)
(N: 00H to FFH)

Note 8-bit timer/event counter 50: P50, PM50
8-bit timer/event counter 51: P51, PM51

Caution Do not write other values to CR5n during operation.

Remark n = 0, 1

Figure 9-12. Square-Wave Output Operation Timing



Note The initial value of TO5n output can be set by bits 2 and 3 (LVR5n, LVS5n) of 8-bit timer mode control register 5n (TMC5n).

9.4.4 PWM output operation

8-bit timer/event counter 5n operates as a PWM output when bit 6 (TMC5n6) of 8-bit timer mode control register 5n (TMC5n) is set to 1.

The duty pulse determined by the value set to 8-bit timer compare register 5n (CR5n) is output from TO5n.

Set the active level width of the PWM pulse to CR5n; the active level can be selected with bit 1 (TMC5n1) of TMC5n.

The count clock can be selected with bits 0 to 2 (TCL5n0 to TCL5n2) of timer clock selection register 5n (TCL5n).

PWM output can be enabled/disabled with bit 0 (TOE5n) of TMC5n.

Caution In PWM mode, make the CR5n rewrite period 3 count clocks of the count clock (clock selected by TCL5n) or more.

Remark n = 0, 1

(1) PWM output basic operation

Setting

<1> Set each register.

- Clear the port output latch (P50 or P51)[Note] and port mode register (PM50 or PM51)[Note] to 0.
- TCL5n: Select the count clock.
- CR5n: Compare value
- TMC5n: Stop the count operation, select PWM mode.
  The timer output F/F is not changed.

TMC5n1	Active Level Selection
0	Active-high
1	Active-low

Timer output enabled
(TMC5n = 01000001B or 01000011B)

<2> The count operation starts when TCE5n = 1.
Clear TCE5n to 0 to stop the count operation.

Note 8-bit timer/event counter 50: P50, PM50
8-bit timer/event counter 51: P51, PM51

PWM output operation

<1> PWM output (output from TO5n) outputs an inactive level until an overflow occurs.

<2> When an overflow occurs, the active level is output. The active level is output until CR5n matches the count value of 8-bit timer counter 5n (TM5n).

<3> After the CR5n matches the count value, the inactive level is output until an overflow occurs again.

<4> Operations <2> and <3> are repeated until the count operation stops.

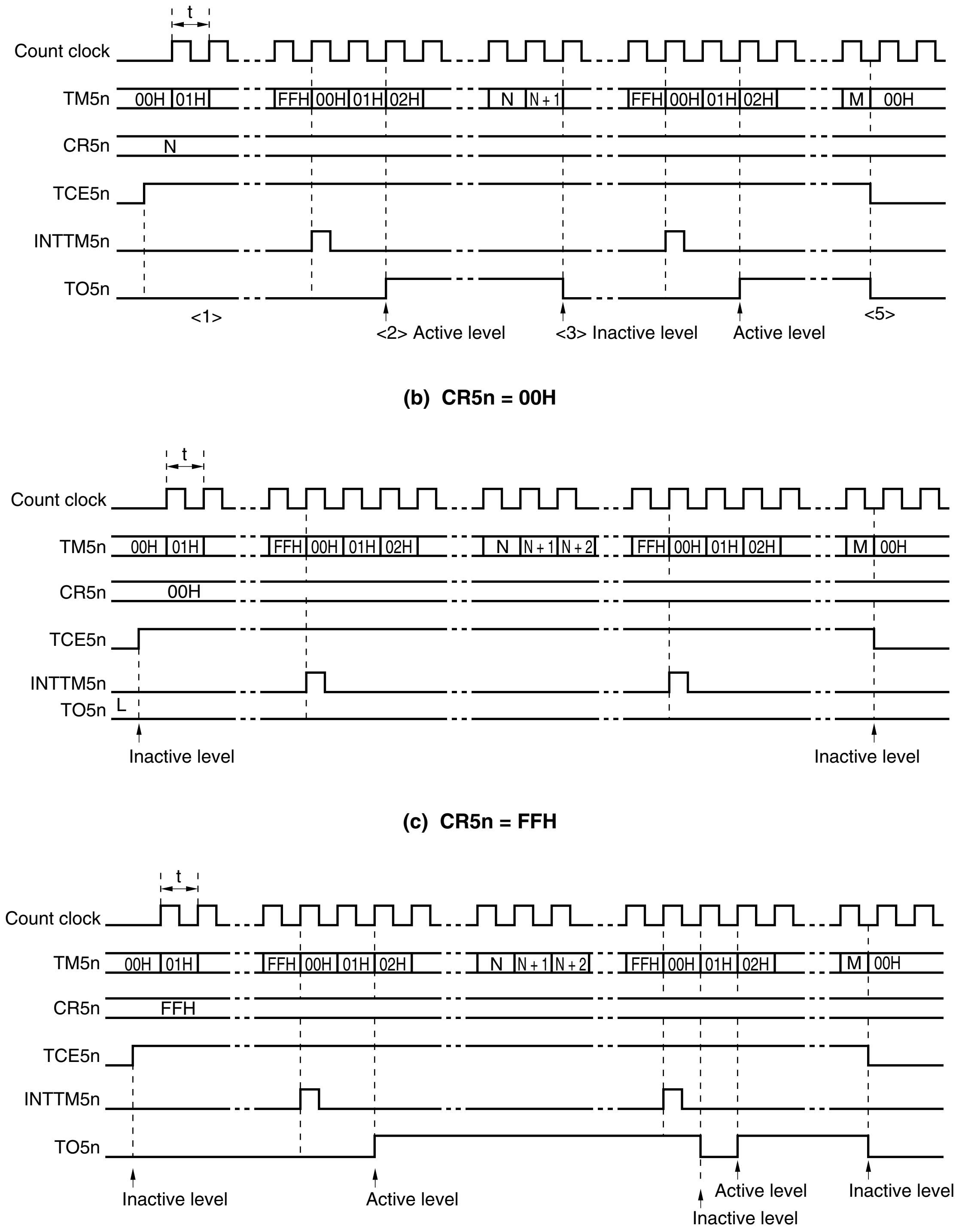
<5> When the count operation is stopped with TCE5n = 0, PWM output becomes inactive.
For details of timing, see **Figures 9-13** and **9-14**.
The cycle, active-level width, and duty are as follows.

- Cycle = 2^8t
- Active-level width = Nt
- Duty = $N/2^8$

(N = 00H to FFH)

Remark n = 0, 1

Figure 9-13. PWM Output Operation Timing



(a) Basic operation (active level = H)

(b) CR5n = 00H

(c) CR5n = FFH

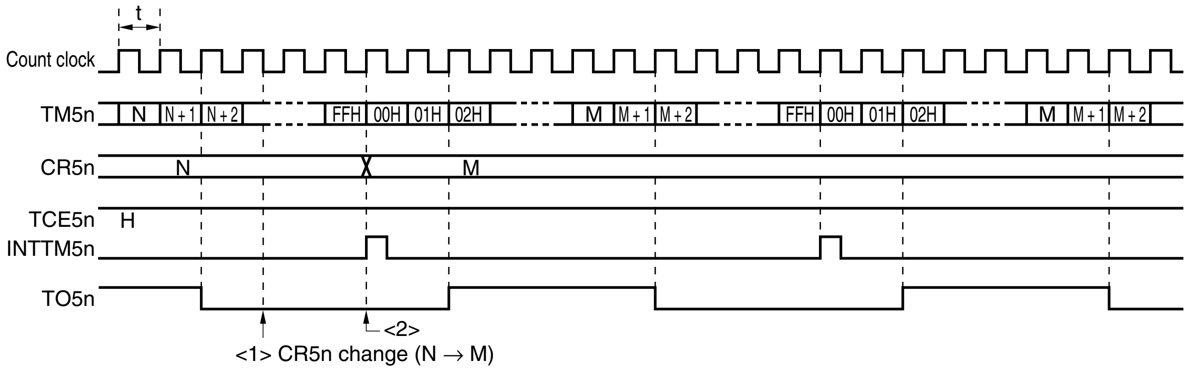
Remarks 1. <1> to <3> and <5> in Figure 9-13 (a) correspond to <1> to <3> and <5> in PWM output operation in **9.4.4 (1) PWM output basic operation**.

2. n = 0, 1

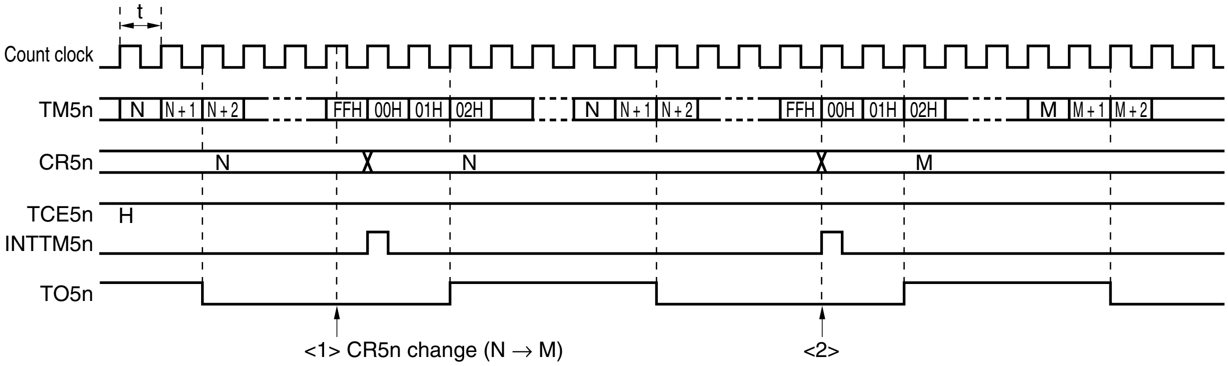
(2) Operation with CR5n changed

Figure 9-14. Timing of Operation with CR5n Changed

(a) CR5n value is changed from N to M before clock rising edge of FFH
→ Value is transferred to CR5n at overflow immediately after change.



(b) CR5n value is changed from N to M after clock rising edge of FFH
→ Value is transferred to CR5n at second overflow.



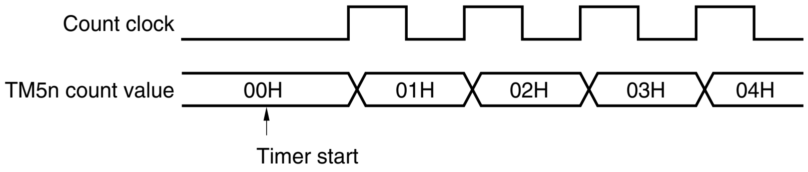
Caution **When reading from CR5n between <1> and <2> in Figure 9-14, the value read differs from the actual value (read value: M, actual value of CR5n: N).**

9.5 Cautions for 8-Bit Timer/Event Counters 50 and 51

(1) Timer start error

An error of up to one clock may occur in the time required for a match signal to be generated after timer start. This is because 8-bit timer counters 50 and 51 (TM50, TM51) are started asynchronously to the count clock.

Figure 9-15. 8-Bit Timer Counter 5n Start Timing



Remark n = 0, 1

CHAPTER 10 8-BIT TIMER H0

10.1 Functions of 8-Bit Timer H0

8-bit timer H0 has the following functions.

- Interval timer
- PWM output mode
- Square-wave output

10.2 Configuration of 8-Bit Timer H0

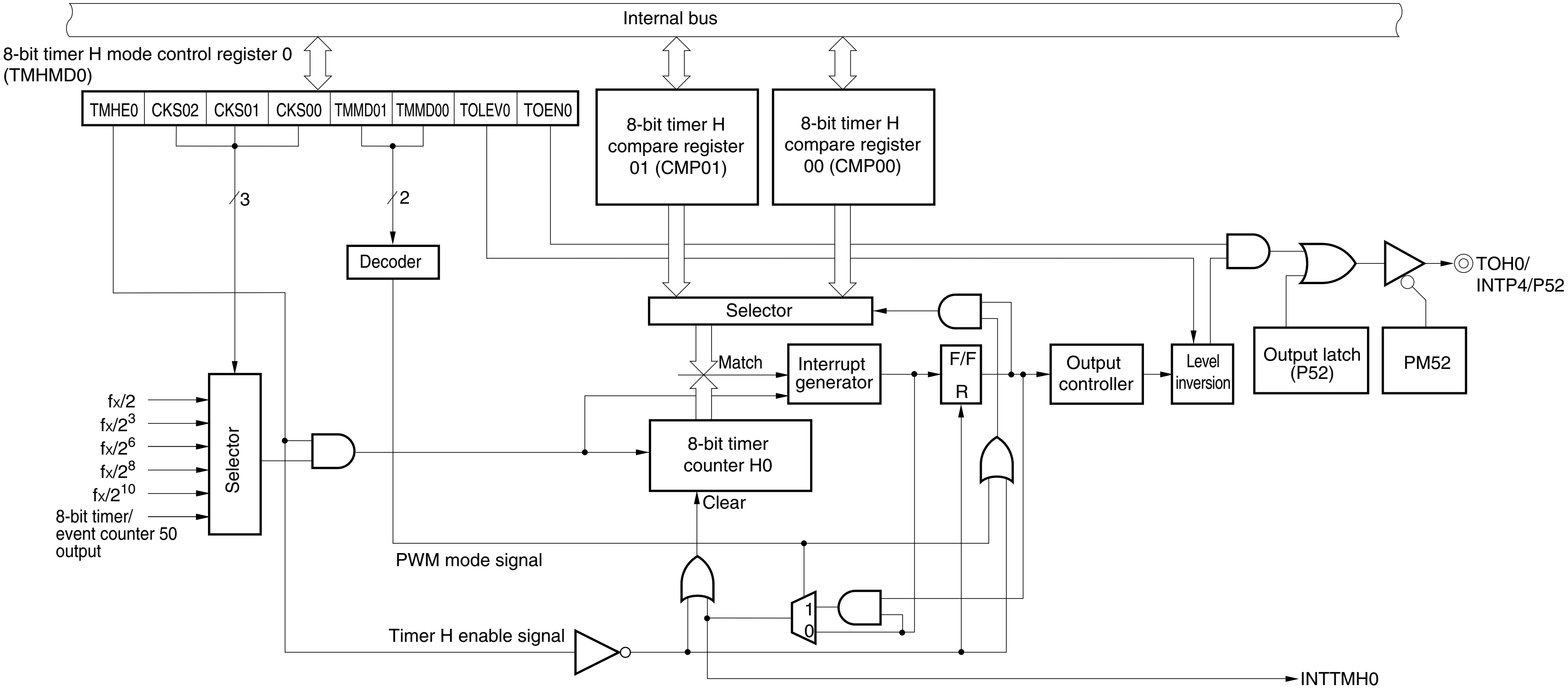
8-bit timer H0 consists of the following hardware.

Table 10-1. Configuration of 8-Bit Timer H0

Item	Configuration
Timer register	8-bit timer counter H0
Registers	8-bit timer H compare register 00 (CMP00) 8-bit timer H compare register 01 (CMP01)
Timer output	TOH0
Control registers	8-bit timer H mode register 0 (TMHMD0) Port mode register 5 (PM5) Port register 5 (P5)

Figures 10-1 shows the block diagram.

Figure 10-1. Block Diagram of 8-Bit Timer H0



(1) 8-bit timer H compare register 00 (CMP00)

This register can be read or written by an 8-bit memory manipulation instruction.

An interrupt request signal (INTTMH0) is generated if the values of the timer counter and CMP00 match. The timer counter value is cleared at the same time.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 10-2. Format of 8-Bit Timer H Compare Register 00 (CMP00)

Address: FF9EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP00								

Caution CMP00 cannot be rewritten during timer count operation.

(2) 8-bit timer H compare register 01 (CMP01)

This register can be read or written by an 8-bit memory manipulation instruction.

This register is used in only PWM output mode.

If the values of the timer counter and CMP01 match, the timer counter value is cleared, but an interrupt request signal (INTTMH0) is not generated.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 10-3. Format of 8-Bit Timer H Compare Register 01 (CMP01)

Address: FF9FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMP01								

CMP01 can be rewritten during timer count operation.

If the CMP01 value is rewritten during timer operation, transferring is performed at the timing at which the counter value and CMP01 value match. If the transfer timing and writing from CPU to CMP01 conflict, transfer is not performed.

Caution In the PWM output mode, be sure to set CMP01 when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP01).

10.3 Registers Controlling 8-Bit Timer H0

The following three registers are used to control 8-bit timer H0.

- 8-bit timer H mode register 0 (TMHMD0)
- Port mode register 5 (PM5)
- Port register 5 (P5)

(1) 8-bit timer H mode register 0 (TMHMD0)

This register controls the mode of timer H.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 10-4. Format of 8-Bit Timer H Mode Register 0 (TMHMD0)

Address: FF2AH After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	<1>	<0>
TMHMD0	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0

TMHE0	Timer operation enable
0	Stops timer count operation (counter is cleared to 0)
1	Enables timer count operation (count operation started by inputting clock)

CKS02	CKS01	CKS00	Count clock (f_{CNT}) selection[Note 1]
0	0	0	$f_X/2$ (10 MHz)
0	0	1	$f_X/2^3$ (2.5 MHz)
0	1	0	$f_X/2^6$ (312.5 kHz)
0	1	1	$f_X/2^8$ (78.125 kHz)
1	0	0	$f_X/2^{10}$ (19.53 kHz)
1	0	1	TM50 output[Note 2]
Other than above			Setting prohibited

TMMD01	TMMD00	Timer operation mode
0	0	Interval timer mode
1	0	PWM output mode
Other than above		Setting prohibited

TOLEV0	Timer output level control (in default mode)
0	Low level
1	High level

TOEN0	Timer output control
0	Disables output
1	Enables output

Notes 1. Be sure to set the count clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Count clock ≤ 10 MHz

2. When the TM50 output is selected as the count clock, observe the following.

- PWM mode (TMC506 = 1)

  Set the clock so that the duty will be 50% and start the operation of 8-bit timer/event counter 50 in advance.

- Clear & start mode entered on match of TM50 and CR50 (TMC506 = 0)

  Enable the timer F/F inversion operation (TMC501 = 1) and start the operation of 8-bit timer/event counter 50 in advance.

It is not necessary to enable the TO50 pin as a timer output pin (bit 0 (TOE50) of the TMC50 register may be 0 or 1), regardless of which mode.

Cautions 1. When the internal oscillation clock is selected as the source clock to the CPU, the clock of the internal oscillator is divided and supplied as the count clock. If the count clock is the internal oscillation clock, the operation of 8-bit timer H0 is not guaranteed.

2. When TMHE0 = 1, setting the other bits of TMHMD0 is prohibited.

3. In the PWM output mode, be sure to set 8-bit timer H compare register 01 (CMP01) when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to CMP01).

Remarks 1. f_X: X1 input clock oscillation frequency

2. Figures in parentheses apply to operation at f_X = 20 MHz

3. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(2) Port mode register 5 (PM5)

This register sets port 5 input/output in 1-bit units.

When using the P52/TOH0/INTP4 pin for timer output, clear PM52 and the output latch of P52 to 0.

PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to FFH.

Figure 10-5. Format of Port Mode Register 5 (PM5)

Address: FF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	P5n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

10.4 Operation of 8-Bit Timer H0

10.4.1 Operation as interval timer/square-wave output

When 8-bit timer counter H0 and compare register 00 (CMP00) match, an interrupt request signal (INTTMH0) is generated and 8-bit timer counter H0 is cleared to 00H.

Compare register 01 (CMP01) is not used in interval timer mode. Since a match of 8-bit timer counter H0 and the CMP01 register is not detected even if the CMP01 register is set, timer output is not affected.

By setting bit 0 (TOEN0) of timer H mode register 0 (TMHMD0) to 1, a square wave of any frequency (duty = 50%) is output from TOH0.

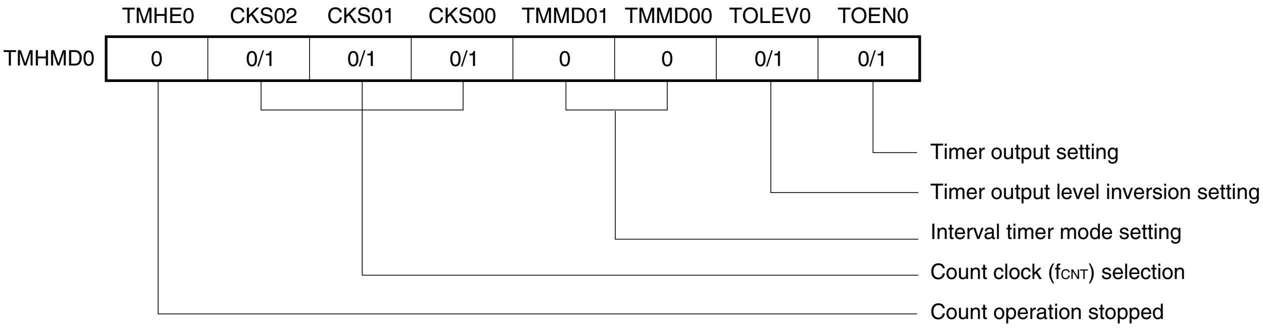
(1) Usage

Generates the INTTMH0 signal repeatedly at the same interval.

<1> Set each register.

Figure 10-6. Register Setting during Interval Timer/Square-Wave Output Operation

(i) Setting timer H mode register 0 (TMHMD0)



(ii) CMP00 register setting

- Compare value (N)

<2> Count operation starts when TMHE0 = 1.

<3> When the values of 8-bit timer counter H0 and the CMP00 register match, the INTTMH0 signal is generated and 8-bit timer counter H0 is cleared to 00H.

Interval time = (N +1)/f_{CNT}

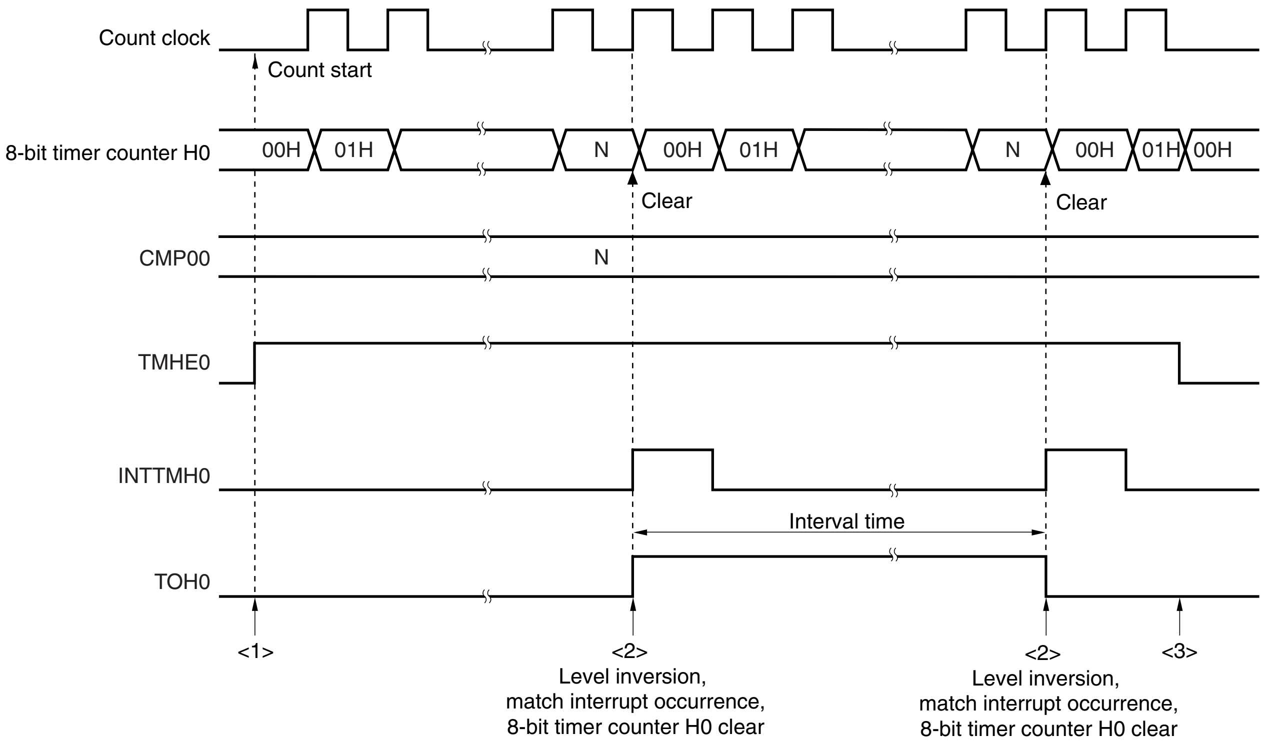
<4> Subsequently, the INTTMH0 signal is generated at the same interval. To stop the count operation, clear TMHE0 to 0.

(2) Timing chart

The timing of the interval timer/square-wave output operation is shown below.

Figure 10-7. Timing of Interval Timer/Square-Wave Output Operation (1/2)

(a) Basic operation



<1> The count operation is enabled by setting the TMHE0 bit to 1. The count clock starts counting no more than 1 clock after the operation is enabled.

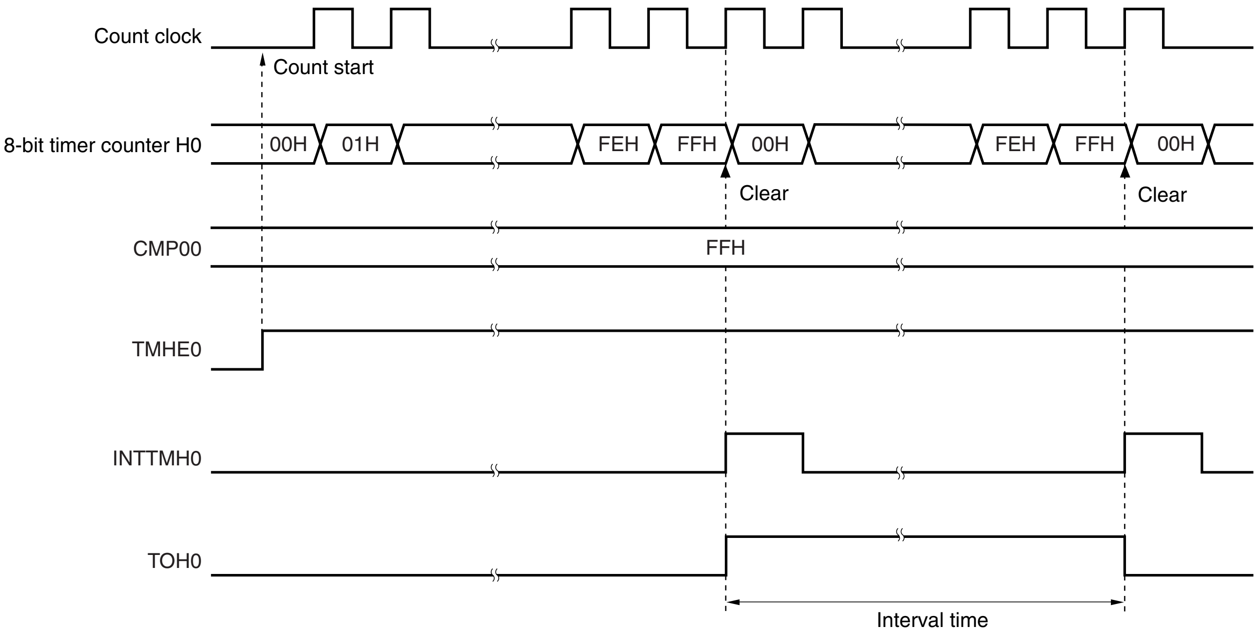
<2> When the values of 8-bit timer counter H0 and the CMP00 register match, the value of 8-bit timer counter H0 is cleared, the TOH0 output level is inverted, and the INTTMH0 signal is output.

<3> The INTTMH0 signal and TOH0 output become inactive by clearing the TMHE0 bit to 0 during timer H0 operation. If these are inactive from the first, the level is retained.

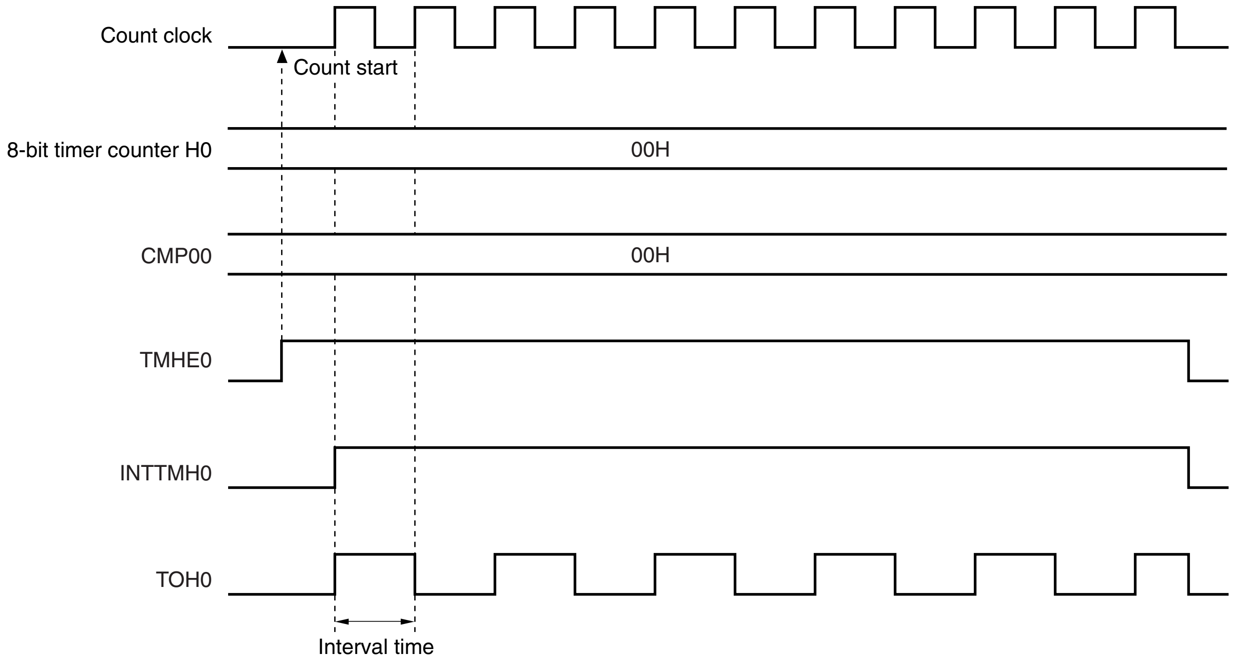
Remark N = 01H to FEH

Figure 10-7. Timing of Interval Timer/Square-Wave Output Operation (2/2)

(b) Operation when CMP00 = FFH



(c) Operation when CMP00 = 00H



10.4.2 Operation as PWM output mode

In PWM output mode, a pulse with an arbitrary duty and arbitrary cycle can be output.

8-bit timer compare register 00 (CMP00) controls the cycle of timer output (TOH0). Rewriting the CMP00 register during timer operation is prohibited.

8-bit timer compare register 01 (CMP01) controls the duty of timer output (TOH0). Rewriting the CMP01 register during timer operation is possible.

The operation in PWM output mode is as follows.

TOH0 output becomes active and 8-bit timer counter H0 is cleared to 0 when 8-bit timer counter H0 and the CMP00 register match after the timer count is started. TOH0 output becomes inactive when 8-bit timer counter H0 and the CMP01 register match.

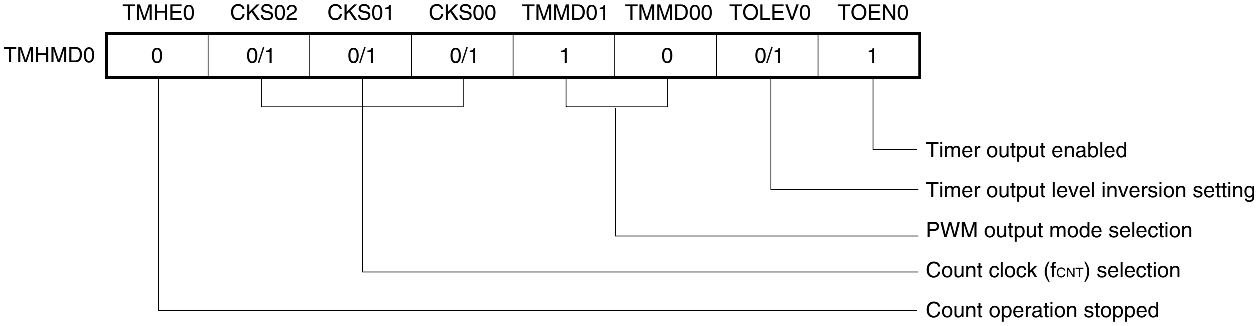
(1) Usage

In PWM output mode, a pulse for which an arbitrary duty and arbitrary cycle can be set is output.

<1> Set each register.

Figure 10-8. Register Setting in PWM Output Mode

(i) Setting timer H mode register 0 (TMHMD0)



	TMHE0	CKS02	CKS01	CKS00	TMMD01	TMMD00	TOLEV0	TOEN0
TMHMD0	0	0/1	0/1	0/1	1	0	0/1	1

Timer output enabled
Timer output level inversion setting
PWM output mode selection
Count clock (f_{CNT}) selection
Count operation stopped

(ii) Setting CMP00 register

- Compare value (N): Cycle setting

(iii) Setting CMP01 register

- Compare value (M): Duty setting

Remark 00H ≤ CMP01 (M) < CMP00 (N) ≤ FFH

<2> The count operation starts when TMHE0 = 1.

<3> The CMP00 register is the compare register that is to be compared first after counter operation is enabled. When the values of 8-bit timer counter H0 and the CMP00 register match, 8-bit timer counter H0 is cleared, an interrupt request signal (INTTMH0) is generated, and TOH0 output becomes active. At the same time, the compare register to be compared with 8-bit timer counter H0 is changed from the CMP00 register to the CMP01 register.

<4> When 8-bit timer counter H0 and the CMP01 register match, TOH0 output becomes inactive and the compare register to be compared with 8-bit timer counter H0 is changed from the CMP01 register to the CMP00 register. At this time, 8-bit timer counter H0 is not cleared and the INTTMH0 signal is not generated.

<5> By performing procedures <3> and <4> repeatedly, a pulse with an arbitrary duty can be obtained.

<6> To stop the count operation, set TMHE0 = 0.

If the setting value of the CMP00 register is N, the setting value of the CMP01 register is M, and the count clock frequency is f_{CNT}, the PWM pulse output cycle and duty are as follows.

PWM pulse output cycle = $(N+1)/f_{CNT}$
Duty = Active width : Total width of PWM = $(M + 1) : (N + 1)$

Cautions 1. In PWM output mode, three operation clocks (signal selected using the CKS02 to CKS00 bits of the TMHMD0 register) are required to transfer the CMP01 register value after rewriting the register.

2. Be sure to set the CMP01 register when starting the timer count operation (TMHE0 = 1) after the timer count operation was stopped (TMHE0 = 0) (be sure to set again even if setting the same value to the CMP01 register).

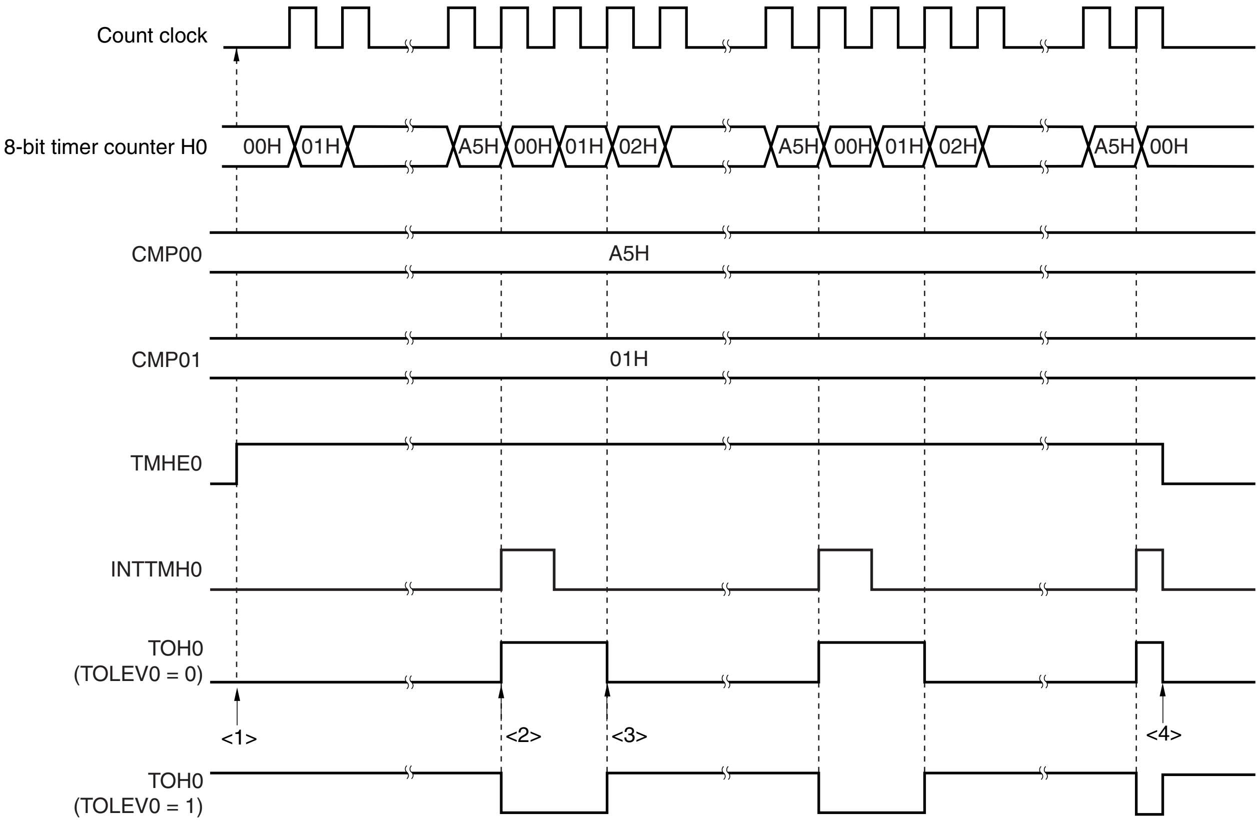
(2) Timing chart

The operation timing in PWM output mode is shown below.

Caution Make sure that the CMP01 register setting value (M) and CMP00 register setting value (N) are within the following range.
00H ≤ CMP01 (M) < CMP00 (N) ≤ FFH

Figure 10-9. Operation Timing in PWM Output Mode (1/4)

(a) Basic operation



<1> The count operation is enabled by setting the TMHE0 bit to 1. Start 8-bit timer counter H0 by masking one count clock to count up. At this time, TOH0 output remains inactive (when TOLEV0 = 0).

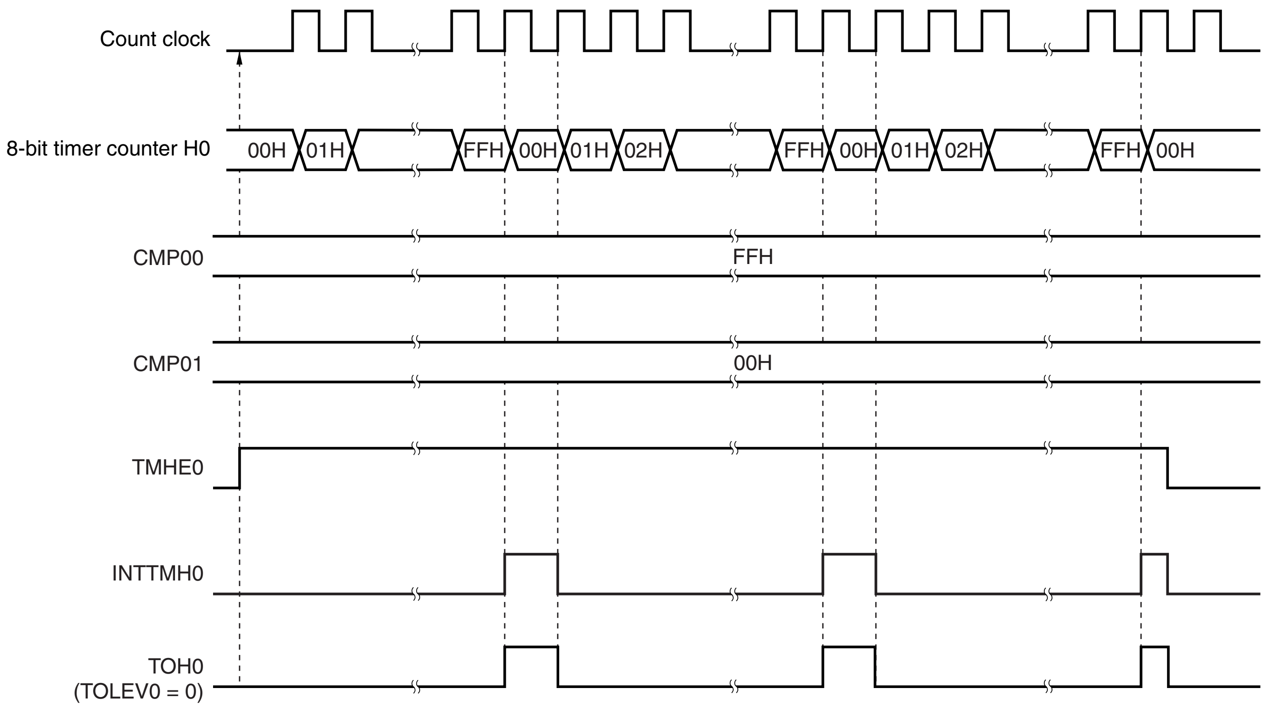
<2> When the values of 8-bit timer counter H0 and the CMP00 register match, the TOH0 output level is inverted, the value of 8-bit timer counter H0 is cleared, and the INTTMH0 signal is output.

<3> When the values of 8-bit timer counter H0 and the CMP01 register match, the level of the TOH0 output is returned. At this time, the 8-bit timer counter value is not cleared and the INTTMH0 signal is not output.

<4> Clearing the TMHE0 bit to 0 during timer H0 operation makes the INTTMH0 signal and TOH0 output inactive.

Figure 10-9. Operation Timing in PWM Output Mode (2/4)

(b) Operation when CMP00 = FFH, CMP01 = 00H



(c) Operation when CMP00 = FFH, CMP01 = FEH

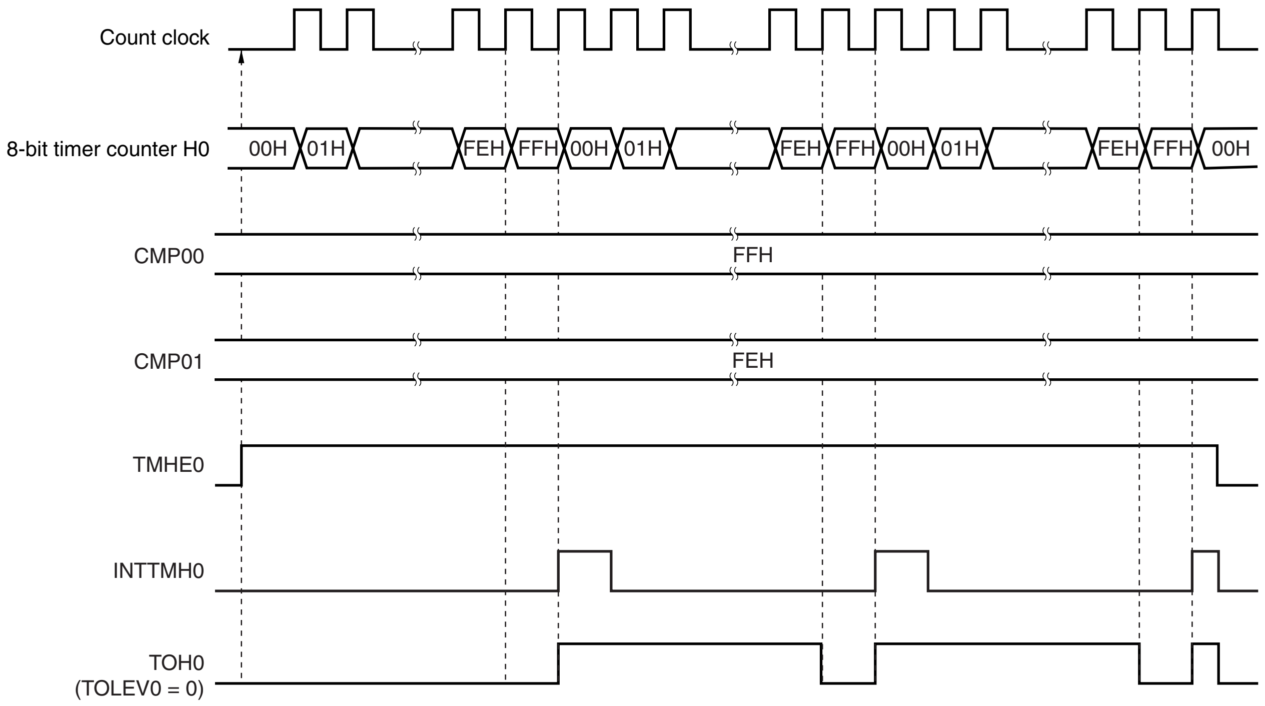


Figure 10-9. Operation Timing in PWM Output Mode (3/4)

(d) Operation when CMP00 = 01H, CMP01 = 00H

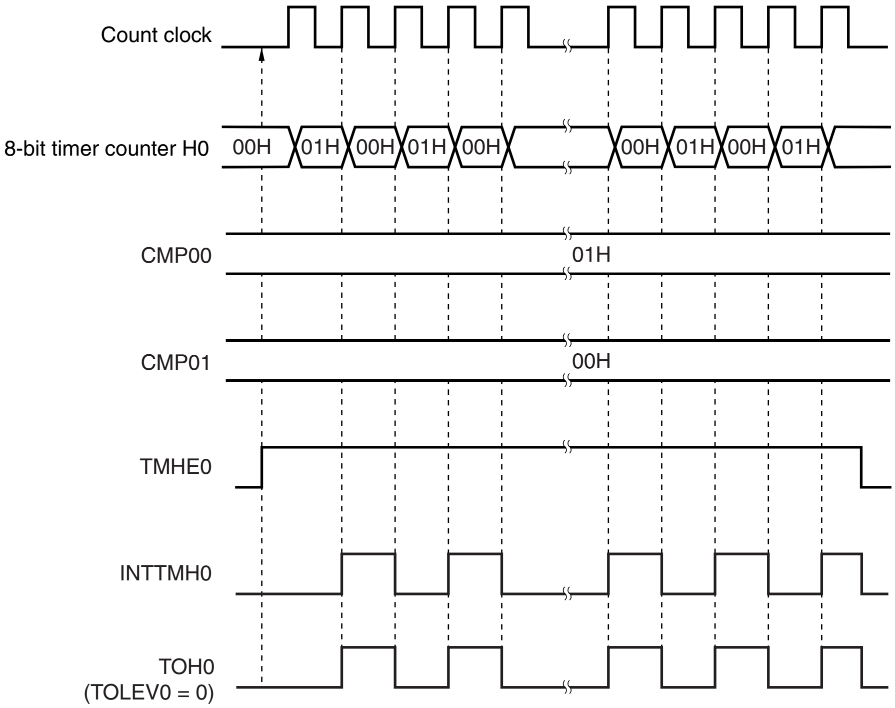
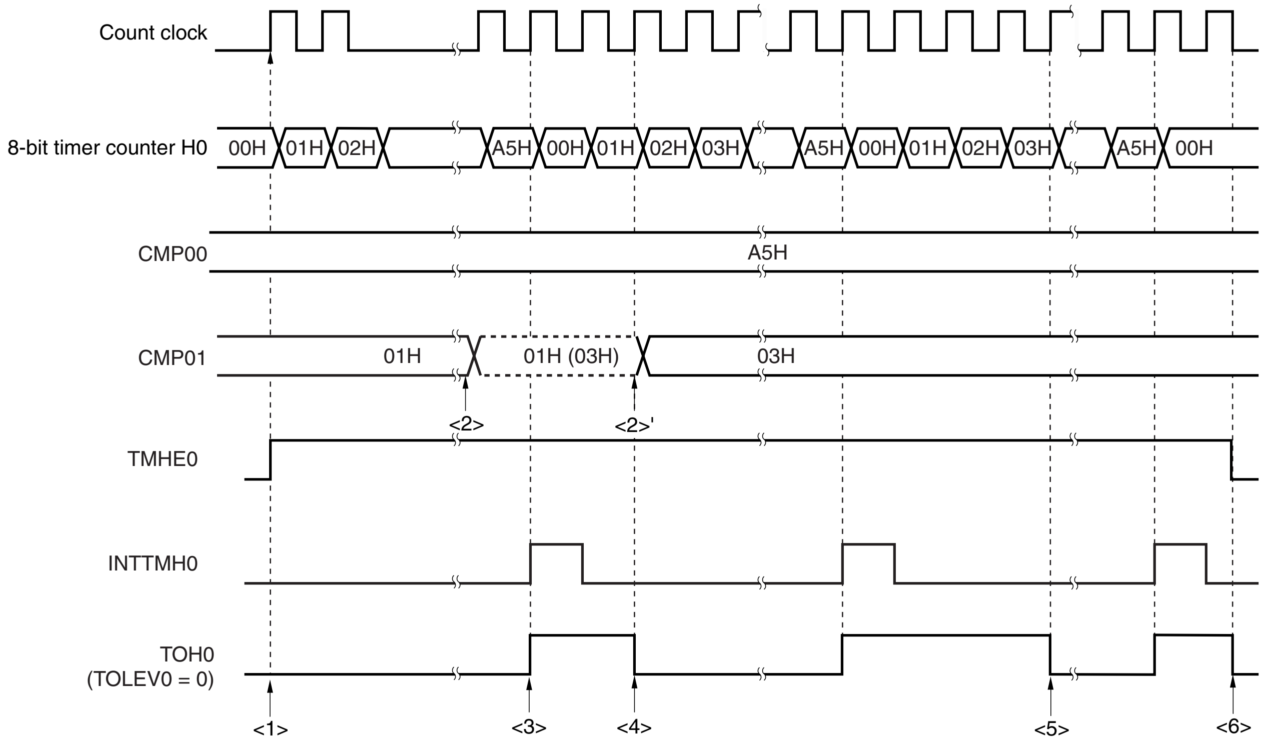


Figure 10-9. Operation Timing in PWM Output Mode (4/4)

(e) Operation by changing CMP01 (CMP01 = 01H → 03H, CMP00 = A5H)



<1> The count operation is enabled by setting TMHE0 = 1. Start 8-bit timer counter H0 by masking one count clock to count up. At this time, the TOH0 output remains inactive (when TOLEV0 = 0).

<2> The CMP01 register value can be changed during timer counter operation. This operation is asynchronous to the count clock.

<3> When the values of 8-bit timer counter H0 and the CMP00 register match, the value of 8-bit timer counter H0 is cleared, the TOH0 output becomes active, and the INTTMH0 signal is output.

<4> If the CMP01 register value is changed, the value is latched and not transferred to the register. When the values of 8-bit timer counter H0 and the CMP01 register before the change match, the value is transferred to the CMP01 register and the CMP01 register value is changed (<2>’).

However, three count clocks or more are required from when the CMP01 register value is changed to when the value is transferred to the register. If a match signal is generated within three count clocks, the changed value cannot be transferred to the register.

<5> When the values of 8-bit timer counter H0 and the CMP01 register after the change match, the TOH0 output becomes inactive. 8-bit timer counter H0 is not cleared and the INTTMH0 signal is not generated.

<6> Clearing the TMHE0 bit to 0 during timer H0 operation makes the INTTMH0 signal and TOH0 output inactive.

CHAPTER 11 WATCHDOG TIMER

11.1 Functions of Watchdog Timer

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

Table 11-1. Loop Detection Time of Watchdog Timer

Loop Detection Time	
During Internal Oscillation Clock Operation	During X1 Input Clock Operation
$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (409.6 μs)
$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (819.2 μs)
$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (1.64 ms)
$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (3.28 ms)
$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (6.55 ms)
$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (13.11 ms)
$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (26.21 ms)
$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (52.43 ms)

Remarks 1. f_R: Internal oscillation clock frequency

2. f_{XP}: X1 input clock oscillation frequency

3. Figures in parentheses apply to operation at f_R = 240 kHz (TYP.), f_{XP} = 20 MHz

The operation mode of the watchdog timer (WDT) is switched according to the option byte setting of the internal oscillator as shown in Table 11-2.

Table 11-2. Option Byte Setting and Watchdog Timer Operation Mode

	Option Byte	
	Internal Oscillator Cannot Be Stopped	Internal Oscillator Can Be Stopped by Software
Watchdog timer clock source	Fixed to f_R[Note 1].	• Selectable by software (f_{XP}, f_R or stopped) • When reset is released: f_R
Operation after reset	Operation starts with the maximum interval ($f_R/2^{18}$).	Operation starts with maximum interval ($f_R/2^{18}$).
Operation mode selection	The overflow time can be changed only once.	The clock selection/overflow time can be changed only once.
Features	The watchdog timer cannot be stopped.	The watchdog timer can be stopped [Note 2].

Notes 1. As long as power is being supplied, internal oscillator oscillation cannot be stopped (except in the reset period).

2. The conditions under which clock supply to the watchdog timer is stopped differ depending on the clock source of the watchdog timer.

<1> If the clock source is f_{XP}, clock supply to the watchdog timer is stopped under the following conditions.

- When f_{XP} is stopped
- In HALT/STOP mode
- During oscillation stabilization time

<2> If the clock source is f_R, clock supply to the watchdog timer is stopped under the following conditions.

- If the CPU clock is f_{XP} and if f_R is stopped by software before execution of the STOP instruction
- In HALT/STOP mode

Remarks 1. f_R: Internal oscillation clock frequency

2. f_{XP}: X1 input clock oscillation frequency

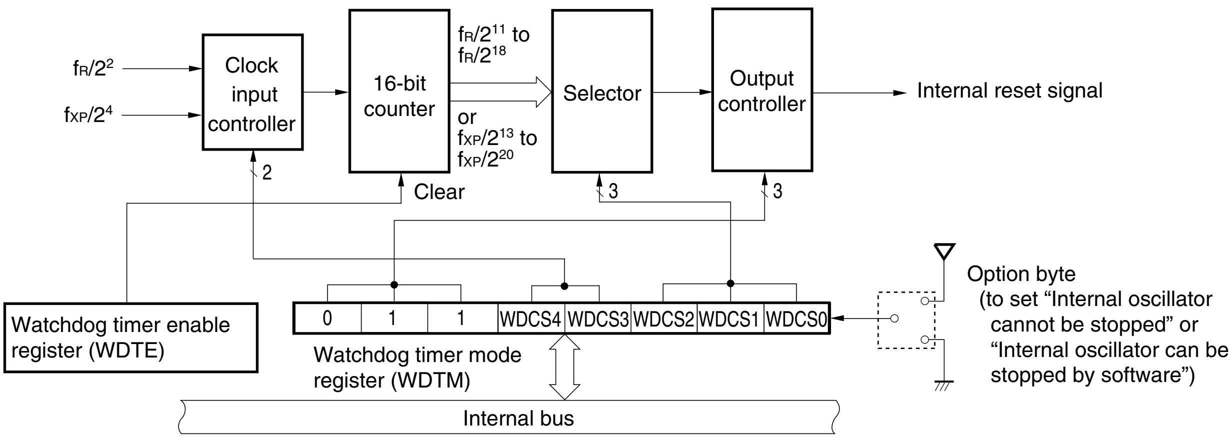
11.2 Configuration of Watchdog Timer

The watchdog timer consists of the following hardware.

Table 11-3. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer mode register (WDTM) Watchdog timer enable register (WDTE)

Figure 11-1. Block Diagram of Watchdog Timer


$f_R/2^2$
$f_{XP}/2^4$
Clock input controller
2
16-bit counter
Clear
$f_R/2^{11}$ to $f_R/2^{18}$ or $f_{XP}/2^{13}$ to $f_{XP}/2^{20}$
Selector
3
Output controller
3
Internal reset signal
Option byte (to set "Internal oscillator cannot be stopped" or "Internal oscillator can be stopped by software")
Watchdog timer enable register (WDTE)
0
1
1
WDCS4
WDCS3
WDCS2
WDCS1
WDCS0
Watchdog timer mode register (WDTM)
Internal bus

11.3 Registers Controlling Watchdog Timer

The watchdog timer is controlled by the following two registers.

- Watchdog timer mode register (WDTM)
- Watchdog timer enable register (WDTE)

(1) Watchdog timer mode register (WDTM)

This register sets the overflow time and operation clock of the watchdog timer.

This register can be set by an 8-bit memory manipulation instruction and can be read many times, but can be written only once after reset is released.

$\overline{\text{RESET}}$ input sets this register to 67H.

Figure 11-2. Format of Watchdog Timer Mode Register (WDTM)

Address: FF98H After reset: 67H R/W

Symbol	7	6	5	4	3	2	1	0
WDTM	0	1	1	WDCS4	WDCS3	WDCS2	WDCS1	WDCS0

WDCS4[Note 1]	WDCS3[Note 1]	Operation clock selection
0	0	Internal oscillation clock (f_R)
0	1	X1 input clock (f_{XP})
1	×	Watchdog timer operation stopped

WDCS2[Note 2]	WDCS1[Note 2]	WDCS0[Note 2]	Overflow time setting	
			During internal oscillation clock operation	During X1 input clock operation
0	0	0	$f_R/2^{11}$ (8.53 ms)	$f_{XP}/2^{13}$ (409.6 μs)
0	0	1	$f_R/2^{12}$ (17.07 ms)	$f_{XP}/2^{14}$ (819.2 μs)
0	1	0	$f_R/2^{13}$ (34.13 ms)	$f_{XP}/2^{15}$ (1.64 ms)
0	1	1	$f_R/2^{14}$ (68.27 ms)	$f_{XP}/2^{16}$ (3.28 ms)
1	0	0	$f_R/2^{15}$ (136.53 ms)	$f_{XP}/2^{17}$ (6.55 ms)
1	0	1	$f_R/2^{16}$ (273.07 ms)	$f_{XP}/2^{18}$ (13.11 ms)
1	1	0	$f_R/2^{17}$ (546.13 ms)	$f_{XP}/2^{19}$ (26.21 ms)
1	1	1	$f_R/2^{18}$ (1.09 s)	$f_{XP}/2^{20}$ (52.43 ms)

Notes 1. If “Internal oscillator cannot be stopped” is specified by an option byte, this cannot be set. The Internal oscillation clock will be selected no matter what value is written.

2. Reset is released at the maximum cycle (WDCS2, 1, 0 = 1, 1, 1).

Cautions 1. If data is written to WDTM, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

2. Set bits 7, 6, and 5 to 0, 1, and 1, respectively (when “Internal oscillator cannot be stopped” is selected by an option byte, other values are ignored).

3. After reset is released, WDTM can be written only once by an 8-bit memory manipulation instruction. If writing attempted a second time, an internal reset signal is generated.

4. WDTM cannot be set by a 1-bit memory manipulation instruction.

Remarks 1. f_R: Internal oscillation clock frequency

2. f_{XP}: X1 input clock oscillation frequency

3. ×: Don’t care

4. Figures in parentheses apply to operation at f_R = 240 kHz (TYP.), f_{XP} = 20 MHz

(2) Watchdog timer enable register (WDTE)

Writing ACH to WDTE clears the watchdog timer counter and starts counting again.

This register can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 9AH.

Figure 11-3. Format of Watchdog Timer Enable Register (WDTE)

Address: FF99H After reset: 9AH R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. If a value other than ACH is written to WDTE, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.

3. The value read from WDTE is 9AH (this differs from the written value (ACH)).

11.4 Operation of Watchdog Timer

11.4.1 Watchdog timer operation when “Internal oscillator cannot be stopped” is selected by option byte

The operation clock of watchdog timer is fixed to the Internal oscillation clock.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1). The watchdog timer operation cannot be stopped.

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Internal oscillation clock
   - Cycle: $f_R/2^{18}$ (1.09 seconds: At operation with f_R = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2].
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. The operation clock (internal oscillation clock) cannot be changed. If any value is written to bits 3 and 4 (WDCS3, WDCS4) of WDTM, it is ignored.

2. As soon as WDTM is written, the counter of the watchdog timer is cleared.

Caution In this mode, operation of the watchdog timer absolutely cannot be stopped even during STOP instruction execution. For 8-bit timer 51 (TM51), a division of the internal oscillation clock can be selected as the count source, so clear the watchdog timer using the interrupt request of TM51 before the watchdog timer overflows after STOP instruction execution. If this processing is not performed, an internal reset signal is generated when the watchdog timer overflows after STOP instruction execution.

11.4.2 Watchdog timer operation when “internal oscillator can be stopped by software” is selected by option byte

The operation clock of the watchdog timer can be selected as either the internal oscillation clock or the X1 input clock.

After reset is released, operation is started at the maximum cycle (bits 2, 1, and 0 (WDCS2, WDCS1, WDCS0) of the watchdog timer mode register (WDTM) = 1, 1, 1).

The following shows the watchdog timer operation after reset release.

1. The status after reset release is as follows.
   - Operation clock: Internal oscillation clock (f_R)
   - Cycle: $f_R/2^{18}$ (1.09 seconds: At operation with f_R = 240 kHz (TYP.))
   - Counting starts
2. The following should be set in the watchdog timer mode register (WDTM) by an 8-bit memory manipulation instruction[Notes 1, 2, 3].
   - Operation clock: Any of the following can be selected using bits 3 and 4 (WDCS3 and WDCS4).
     Internal oscillation clock (f_R)
     X1 input clock (f_{XP})
     Watchdog timer operation stopped
   - Cycle: Set using bits 2 to 0 (WDCS2 to WDCS0)
3. After the above procedures are executed, writing ACH to WDTE clears the count to 0, enabling recounting.

Notes 1. As soon as WDTM is written, the counter of the watchdog timer is cleared.

2. Set bits 7, 6, and 5 to 0, 1, 1, respectively. Do not set the other values.

3. If the watchdog timer is stopped by setting WDCS4 and WDCS3 to 1 and ×, respectively, an internal reset signal is not generated even if the following processing is performed.
- WDTM is written a second time.
- A 1-bit memory manipulation instruction is executed to WDTE.
- A value other than ACH is written to WDTE.

Caution In this mode, watchdog timer operation is stopped during HALT/STOP instruction execution. After HALT/STOP mode is released, counting is started again using the operation clock of the watchdog timer set before HALT/STOP instruction execution by WDTM. At this time, the counter is not cleared to 0 but holds its value.

For the watchdog timer operation during STOP mode and HALT mode in each status, see **11.4.3 Watchdog timer operation in STOP mode** and **11.4.4 Watchdog timer operation in HALT mode**.

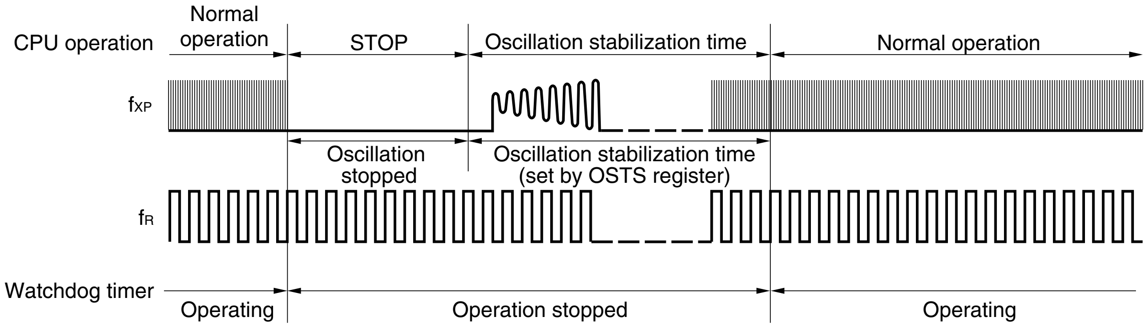
11.4.3 Watchdog timer operation in STOP mode (when "Internal oscillator can be stopped by software" is selected by option byte)

The watchdog timer stops counting during STOP instruction execution regardless of whether the X1 input clock or internal oscillation clock is being used.

(1) When the CPU clock and the watchdog timer operation clock are the X1 input clock (f_{XP}) when the STOP instruction is executed

When STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting stops for the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) and then counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

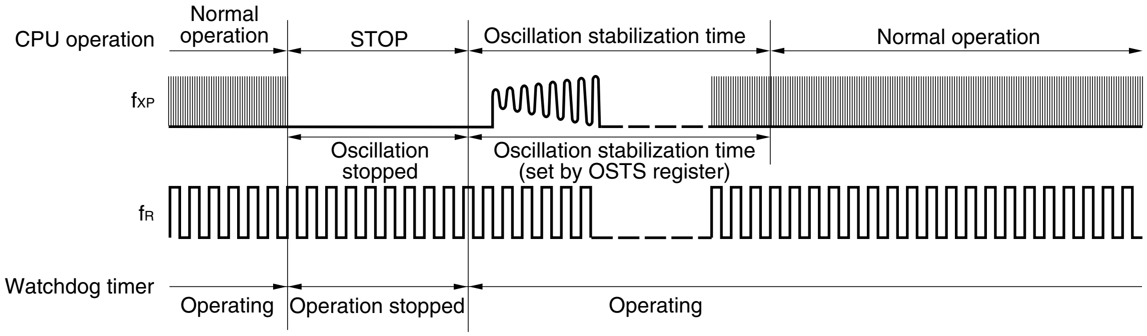
Figure 11-4. Operation in STOP Mode (CPU Clock and WDT Operation Clock: X1 Input Clock)



(2) When the CPU clock is the X1 input clock (f_{XP}) and the watchdog timer operation clock is the internal oscillation clock (f_R) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 11-5. Operation in STOP Mode
(CPU Clock: X1 Input Clock, WDT Operation Clock: Internal Oscillation Clock)



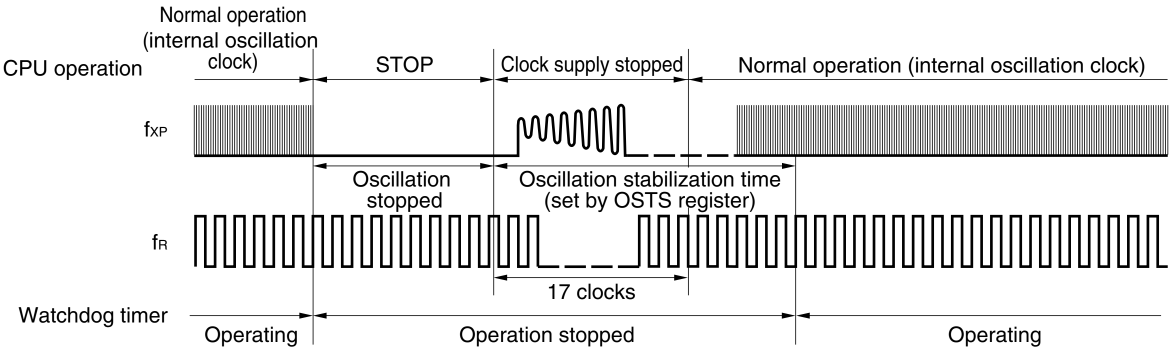
(3) When the CPU clock is the internal oscillation clock (f_R) and the watchdog timer operation clock is the X1 input clock (f_{XP}) when the STOP instruction is executed

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is stopped until the timing of <1> or <2>, whichever is earlier, and then counting is started using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

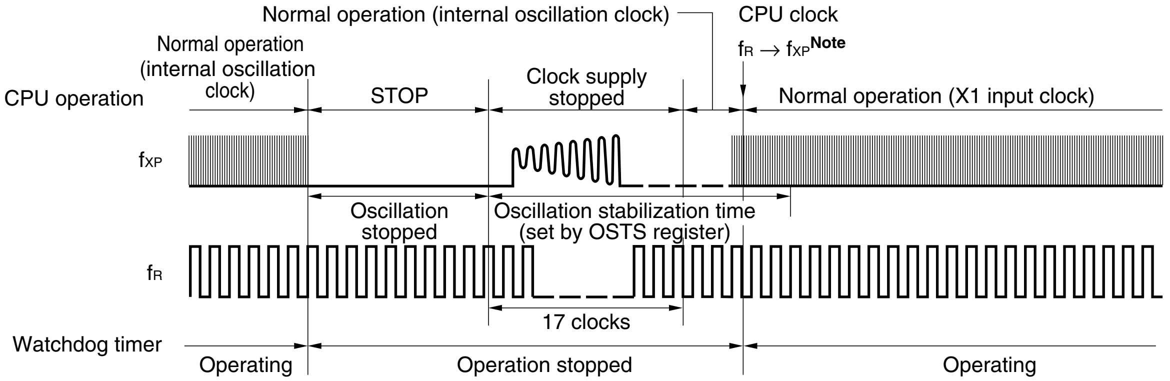
<1> The oscillation stabilization time set by the oscillation stabilization time select register (OSTS) elapses.
<2> The CPU clock is switched to the X1 input clock (f_{XP}).

Figure 11-6. Operation in STOP Mode
(CPU Clock: Internal Oscillation Clock, WDT Operation Clock: X1 Input Clock)

<1> Timing when counting is started after the oscillation stabilization time set by the oscillation stabilization time select register (OSTS) has elapsed


Normal operation (internal oscillation clock)
CPU operation
STOP
Clock supply stopped
Normal operation (internal oscillation clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

<2> Timing when counting is started after the CPU clock is switched to the X1 input clock (f_{XP})

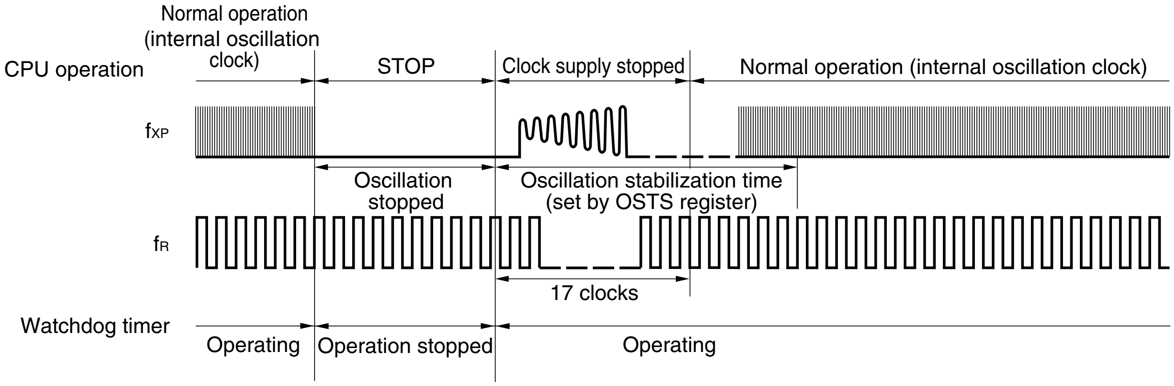

Normal operation (internal oscillation clock)
CPU clock
fR → fXPNote
Normal operation (internal oscillation clock)
CPU operation
STOP
Clock supply stopped
Normal operation (X1 input clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

Note Confirm the oscillation stabilization time of f_{XP} using the oscillation stabilization time counter status register (OSTC).

(4) When CPU clock and watchdog timer operation clock are the internal oscillation clock (f_R) during STOP instruction execution

When the STOP instruction is executed, operation of the watchdog timer is stopped. After STOP mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

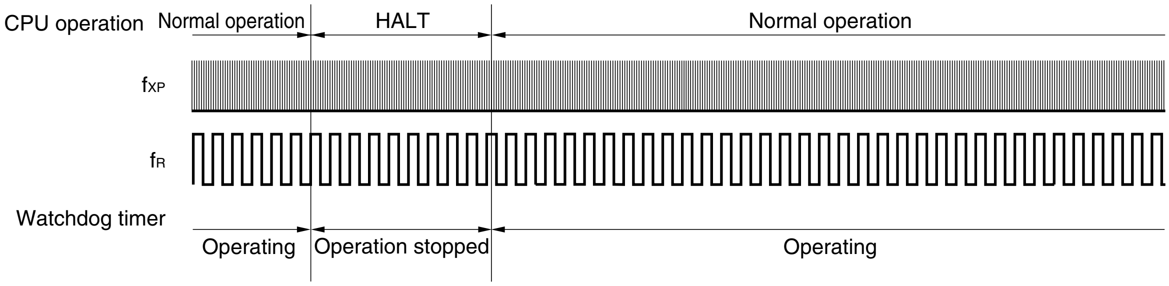
Figure 11-7. Operation in STOP Mode (CPU Clock and WDT Operation Clock: Internal Oscillation Clock)


Normal operation (internal oscillation clock)
CPU operation
STOP
Clock supply stopped
Normal operation (internal oscillation clock)
fXP
Oscillation stopped
Oscillation stabilization time (set by OSTS register)
fR
17 clocks
Watchdog timer
Operating
Operation stopped
Operating

11.4.4 Watchdog timer operation in HALT mode (when “Internal oscillator can be stopped by software” is selected by option byte)

The watchdog timer stops counting during HALT instruction execution regardless of whether the CPU clock is the X1 input clock (f_{XP}) or internal oscillation clock (f_R), or whether the operation clock of the watchdog timer is the X1 input clock (f_{XP}) or internal oscillation clock (f_R). After HALT mode is released, counting is started again using the operation clock before the operation was stopped. At this time, the counter is not cleared to 0 but holds its value.

Figure 11-8. Operation in HALT Mode



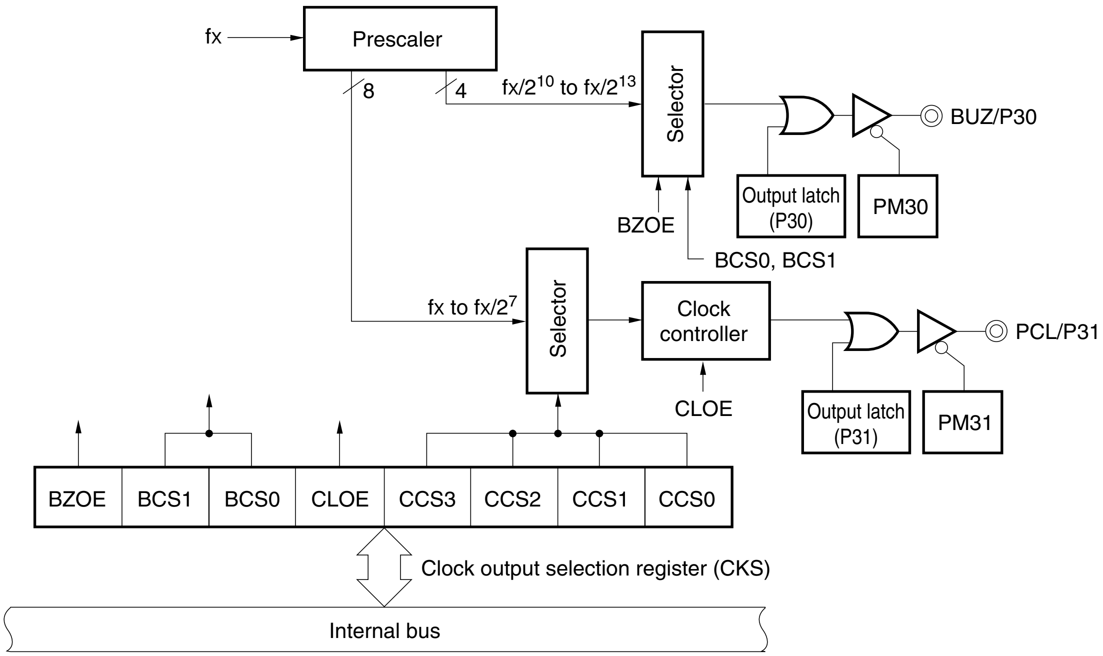
12.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral LSIs. The clock selected with the clock output selection register (CKS) is output.

In addition, the buzzer output is intended for square-wave output of buzzer frequency selected with CKS.

Figure 12-1 shows the block diagram of clock output/buzzer output controller.

Figure 12-1. Block Diagram of Clock Output/Buzzer Output Controller



12.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller consists of the following hardware.

Table 12-1. Clock Output/Buzzer Output Controller Configuration

Item	Configuration
Control registers	Clock output selection register (CKS) Port mode register 3 (PM3) Port register 3 (P3)

12.3 Register Controlling Clock Output/Buzzer Output Controller

The following three registers are used to control the clock output/buzzer output controller.

- Clock output selection register (CKS)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) Clock output selection register (CKS)

This register sets output enable/disable for clock output (PCL) and for the buzzer frequency output (BUZ), and sets the output clock.

CKS is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears CKS to 00H.

Figure 12-2. Format of Clock Output Selection Register (CKS)

Address: FF40H After reset: 00H R/W

Symbol	<7>	6	5	<4>	3	2	1	0
CKS	BZOE	BCS1	BCS0	CLOE	CCS3	CCS2	CCS1	CCS0

BZOE	BUZ output enable/disable specification
0	Clock division circuit operation stopped. BUZ fixed to low level.
1	Clock division circuit operation enabled. BUZ output enabled.

BCS1	BCS0	BUZ output clock selection
0	0	$f_X/2^{10}$ (19.5 kHz)
0	1	$f_X/2^{11}$ (9.77 kHz)
1	0	$f_X/2^{12}$ (4.88 kHz)
1	1	$f_X/2^{13}$ (2.44 kHz)

CLOE	PCL output enable/disable specification
0	Clock division circuit operation stopped. PCL fixed to low level.
1	Clock division circuit operation enabled. PCL output enabled.

CCS3	CCS2	CCS1	CCS0	PCL output clock selection
0	0	0	0	f_X (20 MHz)
0	0	0	1	$f_X/2$ (10 MHz)
0	0	1	0	$f_X/2^2$ (5 MHz)
0	0	1	1	$f_X/2^3$ (2.5 MHz)
0	1	0	0	$f_X/2^4$ (1.25 MHz)
0	1	0	1	$f_X/2^5$ (625 kHz)
0	1	1	0	$f_X/2^6$ (312.5 kHz)
0	1	1	1	$f_X/2^7$ (156.25 kHz)
Other than above				Setting prohibited

Remarks 1. f_X: X1 input clock oscillation frequency

2. Figures in parentheses are for operation with f_X = 20 MHz.

(2) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P31/ PCL pin for clock output and the P30/BUZ pin for buzzer output, clear PM31, PM30 and the output latch of P31, P30 to 0.

PM3 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets PM3 to FFH.

Figure 12-3. Format of Port Mode Register 3 (PM3)

Address: FF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 3)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

12.4 Clock Output/Buzzer Output Controller Operations

12.4.1 Clock output operation

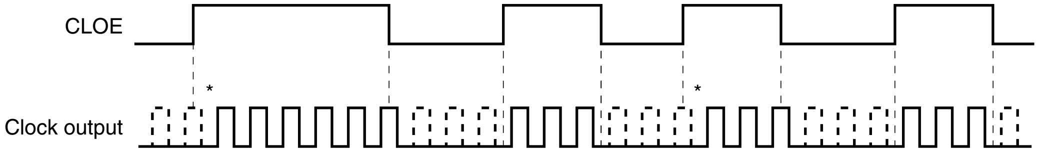
The clock pulse is output as the following procedure.

<1> Select the clock pulse output frequency with bits 0 to 3 (CCS0 to CCS3) of the clock output selection register (CKS) (clock pulse output in disabled status).

<2> Set bit 4 (CLOE) of CKS to 1 to enable clock output.

Remark The clock output controller is designed not to output pulses with a small width during output enable/disable switching of the clock output. As shown in Figure 12-4, be sure to start output from the low period of the clock (marked with * in the figure). When stopping output, do so after securing high level of the clock.

Figure 12-4. Remote Control Output Application Example



12.4.2 Operation as buzzer output

The buzzer frequency is output as the following procedure.

<1> Select the buzzer output frequency with bits 5 and 6 (BCS0, BCS1) of the clock output selection register (CKS) (buzzer output in disabled status).

<2> Set bit 7 (BZOE) of CKS to 1 to enable buzzer output.

CHAPTER 13 REAL-TIME OUTPUT PORT

13.1 Function of Real-Time Output Port

Data set previously in the real-time output buffer register can be transferred to the output latch by hardware concurrently with timer interrupts or external interrupt request generation, then output externally. This is called the real-time output function. The pins that output data externally are called real-time output ports.

By using the real-time output port, it is possible to output a signal with no jitter. Therefore, this is most suitable for applications where an arbitrary pattern is output at an arbitrary interval (open-loop control of a stepper motor, etc.).

Also, it is possible to perform PWM modulation at a specified pin for the output pattern.

The μPD78F0714 has the following 2 channels of real-time output ports on chip. It is possible to specify the real-time output port in 1-bit units.

- 8 bits × 1, or 4 bits × 2 … Real-time output port 0
- 6 bits × 1, or 4 bits × 1 … Real-time output port 1

13.2 Configuration of Real-Time Output Port

A real-time output port includes the following hardware.

Table 13-1. Configuration of Real-Time Output Port

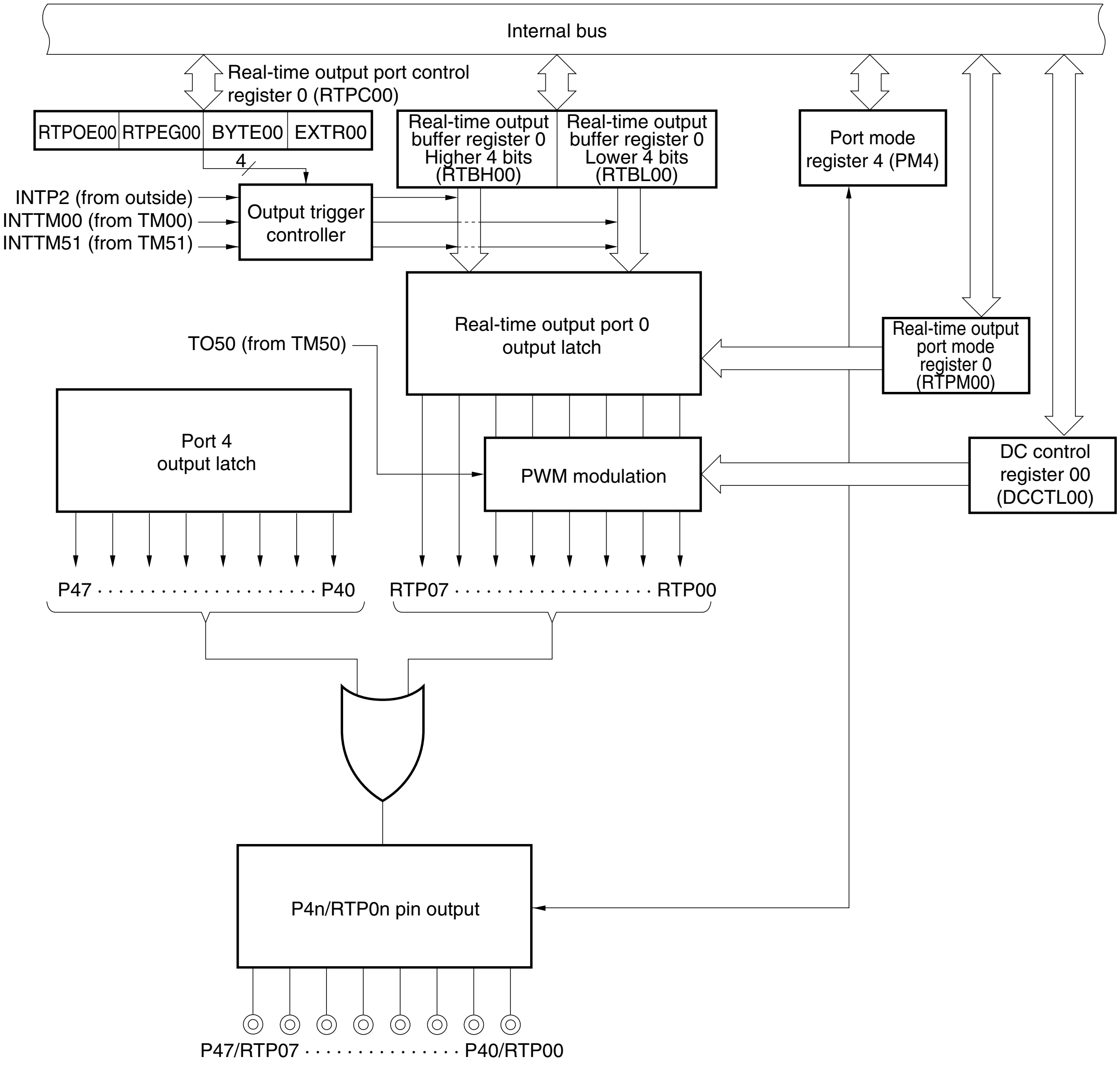
Item	Configuration
Register	Real-time output buffer register n (RTBL0n, RTBH0n)
Control registers	Port mode register 4 (PM4) Real-time output port mode register n (RTPM0n) Real-time output port control register n (RTPC0n) DC control register 0n (DCCTL0n)

n = 0, 1.

<R>

Figure 13-1. Block Diagram of Real-Time Output Port (1/2)

(a) Real-time output port 0 (8 bits × 1, or 4 bits × 2)

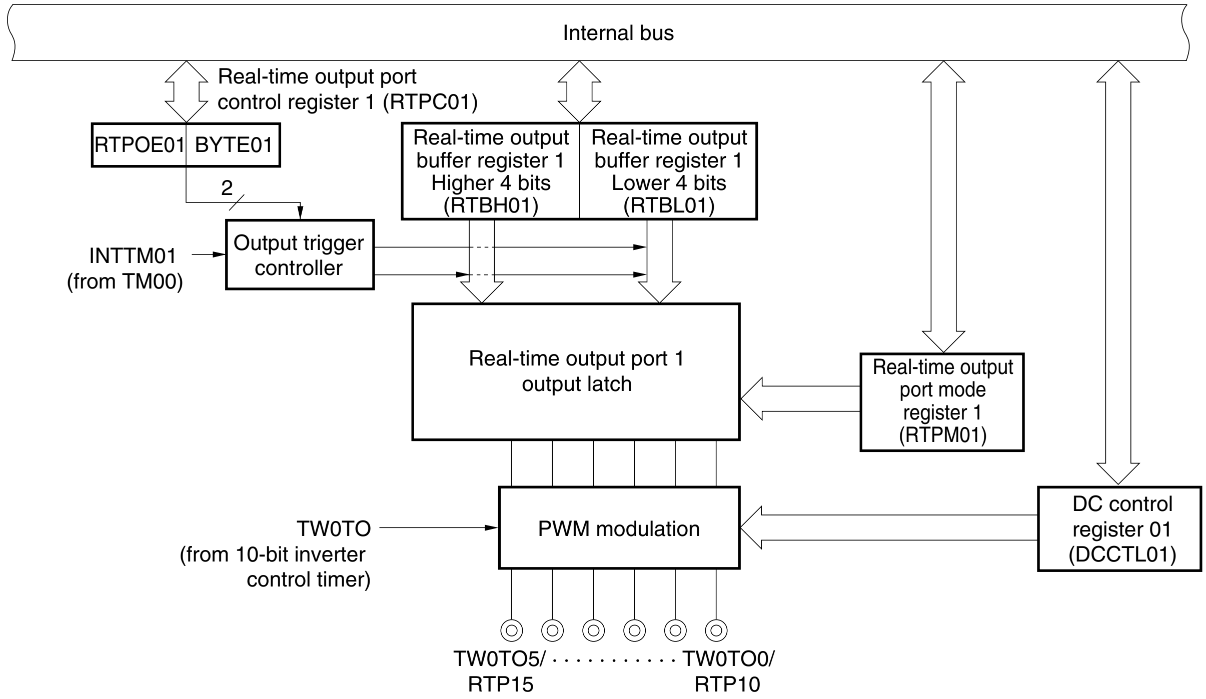

Internal bus
Real-time output port control register 0 (RTPC00)
RTPOE00
RTPEG00
BYTE00
EXTR00
4
Real-time output buffer register 0 Higher 4 bits (RTBH00)
Real-time output buffer register 0 Lower 4 bits (RTBL00)
Port mode register 4 (PM4)
INTP2 (from outside)
INTTM00 (from TM00)
INTTM51 (from TM51)
Output trigger controller
Real-time output port 0 output latch
TO50 (from TM50)
Real-time output port mode register 0 (RTPM00)
Port 4 output latch
PWM modulation
DC control register 00 (DCCTL00)
P47 · P40
RTP07 · · · · · · · · · · · · · · · · · · · RTP00
P4n/RTP0n pin output
P47/RTP07 · · · · · · · · · · · · · · · P40/RTP00

Remark n = 0 to 7

Figure 13-1. Block Diagram of Real-Time Output Port (2/2)

(b) Real-time output port 1 (6 bits × 1, or 4 bits × 1)

<R>



Remark n = 0 to 5

(1) Real-time output buffer register 0 (RTBL00, RTBH00)

This register consists of two 4-bit registers that hold output data in advance.

The addresses of RTBL00 and RTBH00 are mapped individually in the special function register (SFR) area as shown in Figure 13-2.

When specifying 4 bits × 2 channels as the operation mode, data is set individually in RTBL00 and RTBH00.

The data of both RTBL00 and RTBH00 can be read all at once regardless of which address is specified.

When specifying 8 bits × 1 channel as the operation mode, data is set to both RTBL00 and RTBH00 by writing 8-bit data to either RTBL00 or RTBH00. The data of both RTBL00 and RTBH00 can be read all at once regardless of which address is specified.

Figure 13-2 shows the configuration of RTBL00 and RTBH00, and Table 13-2 shows operations during manipulation of RTBL00 and RTBH00.

Figure 13-2. Configuration of Real-Time Output Buffer Register 0

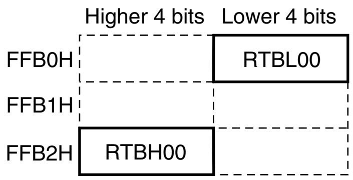


Table 13-2. Operation During Manipulation of Real-Time Output Buffer Register 0

<R>

Operating Mode	Register to Be Manipulated	Reading		Writing[Note]	
		Higher 4 Bits	Lower 4 Bits	Higher 4 Bits	Lower 4 Bits
4 bits × 2 channels	RTBL00	RTBH00	RTBL00	Invalid	RTBL00
	RTBH00	RTBH00	RTBL00	RTBH00	Invalid
8 bits × 1 channel	RTBL00	RTBH00	RTBL00	RTBH00	RTBL00
	RTBH00	RTBH00	RTBL00	RTBH00	RTBL00

Note After setting data in the real-time output port, output data should be set in RTBL00 and RTBH00 by the time a real-time output trigger is generated.

(2) Real-time output buffer register 1 (RTBL01, RTBH01)

This register consists of two 4-bit Note registers that hold output data in advance.

The addresses of RTBL01 and RTBH01 are mapped individually in the special function register (SFR) area as shown in Figure 13-3.

When specifying 4 bits × 1 channel as the operation mode, data is set in RTBL01.

When specifying 6 bits × 1 channel as the operation mode, data is set to both RTBL01 and RTBH01 by writing 6-bit data to either RTBL01 or RTBH01. The data of both RTBL01 and RTBH01 can be read all at once regardless of which address is specified.

Figure 13-3 shows the configuration of RTBL01 and RTBH01, and Table 13-3 shows operations during manipulation of RTBL01 and RTBH01.

Note For RTBH01, only 2 of the 4 bits are valid.

Figure 13-3. Configuration of Real-Time Output Buffer Register 1

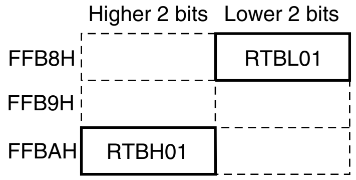


Table 13-3. Operation During Manipulation of Real-Time Output Buffer Register 1

<R>

Operating Mode	Register to Be Manipulated	Reading		Writing[Note]	
		Higher 2 Bits	Lower 4 Bits	Higher 2 Bits	Lower 4 Bits
4 bits × 1 channel	RTBL01	Invalid	RTBL01	Invalid	RTBL01
6 bits × 1 channel	RTBL01	RTBH01	RTBL01	RTBH01	RTBL01
	RTBH01	RTBH01	RTBL01	RTBH01	RTBL01

Note After setting data in the real-time output port, output data should be set in RTBL01 and RTBH01 by the time a real-time output trigger is generated.

13.3 Registers Controlling Real-Time Output Port

The following seven types of registers control the real-time output ports.

- Port mode register 4 (PM4)
- Real-time output port mode register 0, 1 (RTPM00, RTPM01)
- Real-time output port control register 0, 1 (RTPC00, RTPC01)
- DC control register 00, 01 (DCCTL00, DCCTL01)

(1) Port mode register 4 (PM4)

This register sets the input/output mode of port 4 pins (P40 to P47) that function alternately as real-time output pins (RTP00 to RTP07). To use port 4 as a real-time output port, the input/output mode of the port pins used as real-time output port pins must be set in the output mode (PM4n = 0: n = 0 to 7).

PM4 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to FFH.

Figure 13-4. Format of Port Mode Register 4

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FF24H	FFH	R/W

PM4n	P4n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

(2) Real-time output port mode register 0 (RTPM00)

This register sets the real-time output port mode or port mode in 1-bit units.

The outputs to be set are RTP00 to RTP07.

RTPM00 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 13-5. Format of Real-Time Output Port Mode Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RTPM00	RTPM007	RTPM006	RTPM005	RTPM004	RTPM003	RTPM002	RTPM001	RTPM000	FFB4H	00H	R/W

RTPM00n	Real-time output port selection (n = 0 to 7)
0	Port mode
1	Real-time output port mode

Caution When using a port as a real-time output port, set the port in the output mode (by clearing the corresponding bit of port mode register 4 (PM4) to 0).
RTPM007 RTPM006 RTPM005 RTPM004 RTPM002 RTPM001RTPM000 RTPM00 RTPM00n Real-time output port selection (n = 0 to 7)

(3) Real-time output port mode register 1 (RTPM01)

This register is used to set the real-time output port mode in advance, in 1-bit units.

The outputs to be set are RTP10 to RTP15.

RTPM01 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 13-6. Format of Real-Time Output Port Mode Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RTPM01	0	0	RTPM015	RTPM014	RTPM013	RTPM012	RTPM011	RTPM010	FFBCH	00H	R/W

	RTPM01n	Real-time output port selection (n = 0 to 5)
<R>	0	Real-time output buffer is disabled
<R>	1	Real-time output buffer is enabled

Caution Be sure to set bit 6 and 7 of RTPM01 to 0.

Remark When using as a real-time output port, RTP10 to RTP15 become the outputs.

(4) Real-time output port control register 0 (RTPC00)

This register is used to set the operation mode, output trigger and operation enable/disable of the real-time output port.

The outputs to be set are RTP00 to RTP07.

The relationship between the operation mode of the real-time output port and output trigger is as shown in Table 13-4.

RTPC00 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 13-7. Format of Real-Time Output Port Control Register 0

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RTPC00	RTPOE00	RTPEG00	BYTE00	EXTR00	0	0	0	0	FFB5H	00H	R/W

RTPOE00	Real-time output port operation control
0	Disables operation[Note]
1	Enables operation

RTPEG00	INTP2 valid edge specification
0	Falling edge
1	Rising edge

BYTE00	Real-time output port operation mode
0	4 bits × 2 channels
1	8 bits × 1 channel

EXTR00	Real-time output control by INTP2
0	INTP2 not used as real-time output trigger.
1	INTP2 used as real-time output trigger.

Note When RTPM00n (bit n (n = 0 to 7) of real-time output port mode register 0 (RTPM00)) is 1, INV00 (bit 4 of DC control register 00 (DCCTL00)) is 0, and real-time output operation is disabled (RTPOE00 = 0), RTP00 to RTP07 output "0".

Table 13-4. Real-Time Output Port Operation Mode and Output Trigger

BYTE00	EXTR00	Operation Mode	RTBH00 → Port Output	RTBL00 → Port Output
0	0	4 bits × 2 channels	INTTM51	INTTM00
0	1		INTTM00	INTP2
1	0	8 bits × 1 channel	INTTM00	
1	1		INTP2	

(5) Real-time output port control register 1 (RTPC01)

This register is used to set the operation mode, and enabling or disabling operation of the real-time output port.

The outputs to be set are RTP10 to RTP15.

The relationship between the operation mode of the real-time output port and output trigger is as shown in Table 13-5.

RTPC01 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 13-8. Format of Real-Time Output Port Control Register 1

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
RTPC01	RTPOE01	0	BYTE01	0	0	0	0	0	FFBDH	00H	R/W

RTPOE01	Real-time output port operation control
0	Disables operation[Note]
1	Enables operation

BYTE01	Real-time output port operation mode
0	4 bits × 1 channel
1	6 bits × 1 channel

Note When RTPM01n (bit n (n = 0 to 5) of real-time output port mode register 1 (RTPM01)) is 1, INV01 (bit 4 of DC control register 01 (DCCTL01)) is 0, and real-time output operation is disabled (RTPOE01 = 0), RTP10 to RTP15 output “0”.

Table 13-5. Real-Time Output Port Operation Mode and Output Trigger

BYTE01	Operation Mode	RTBH01 → Port Output	RTBL01 → Port Output
0	4 bits × 1 channel	–	INTTM01
1	6 bits × 1 channel	INTTM01	

(6) DC control register 00 (DCCTL00)

This register is used to enable/disable PWM modulation, and enable/disable inversion of the output waveform of the real-time output port.

The outputs to be set are RTP00 to RTP07.

DCCTL00 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 13-9. Format of DC Control Register 00

Address: FF28H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DCCTL00	DCEN00	PWMCH00	PWMCL00	INV00	0	0	0	0

DCEN00	Output operation specification
0	RTP output
1	PWM modulated RTP output[Note]

PWMCH00	PWM modulation specification (RTP00, RTP02, RTP04 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

PWMCL00	PWM modulation specification (RTP01, RTP03, RTP05 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

INV00	Output waveform specification
0	Inversion disabled
1	Inversion enabled

Note The PWM signal uses the TO50 output.

Remarks **1.** The outputs to be set are RTP00 to RTP07.

2. The PWMCH00, PWMCL00, and INV00 settings are valid only when DCEN00 = 1.

(7) DC control register 01 (DCCTL01)

This register is used to enable/disable PWM modulation, and enable/disable inversion of the output waveform of the real-time output port.

The outputs to be set are RTP10 to RTP15.

DCCTL01 is set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 13-10. Format of DC Control Register 01

Address: FF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DCCTL01	DCEN01	PWMCH01	PWMCL01	INV01	0	0	0	0

DCEN01	Output operation specification
0	Inverter timer output (RTP10 to RTP15)
1	PWM modulated RTP output[Note]

PWMCH01	PWM modulation specification (RTP10, RTP12, RTP14 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

PWMCL01	PWM modulation specification (RTP11, RTP13, RTP15 output specification)
0	PWM modulation disabled
1	PWM modulation enabled

INV01	Output waveform specification
0	Inversion disabled
1	Inversion enabled

Note The PWM signal uses the inverter timer outputs (TW0TO0 to TW0TO5).

Remarks **1.** The outputs to be set are RTP10 to RTP15.

2. The PWMCH01, PWMCL01, and INV01 settings are valid only when DCEN01 = 1.

13.4 Operation of Real-Time Output Port

(1) Using RTP00 to RTP07 as the real-time output port ..... Real-time output port 0 (8 bits × 1, or 4 bits × 2)

When bit 7 (RTPOE00) of real-time output port control register 0 (RTPC00) is 1, and real-time output operation is enabled, the data in real-time output buffer register 0 (RTBH00, RTBL00) is transferred to the output latch in synchronization with the generation of the selected transfer trigger (set by EXTR00 and BYTE00). Of the transferred data, only the data of the bit specified for the real-time output port by setting real-time output port mode register 0 (RTPM00) is output from bits RTP00 to RTP07. The ports specified as port mode by RTPM00 can be used as general-purpose input/output ports.

The operation mode can be selected as 8 bits × 1, or 4 bits × 2, by setting EXTR00 and BYTE00. By setting INV00, it is possible to invert the output waveform. Also, by setting PWMCL00 and PWMCH00, it is possible to perform PWM modulation of the output pattern.

If real-time output was disabled (RTPOE00 = 0) when RTPM00n = 1 and INV00 = 0, then RTP00 to RTP07 output 0.

The relationship between the settings for each bit of the control register and the real-time output is shown in Table 13-6, and an example of the operation timing is shown in Figure 13-11.

Remark	EXTR00:	Bit 4 of real-time output port control register 0 (RTPC00)
	BYTE00:	Bit 5 of real-time output port control register 0 (RTPC00)
	INV00:	Bit 4 of DC control register 00 (DCCTL00)
	PWMCL00:	Bit 5 of DC control register 00 (DCCTL00)
	PWMCH00:	Bit 6 of DC control register 00 (DCCTL00)
	RTPM00n:	Bit n (n = 0 to 7) of real-time output port mode register 0 (RTPM00).

Table 13-6. Relationship Between Settings of Each Bit of Control Register and Real-Time Output

PM4n	P4n	DCEN00	INV00	PWMCH00/ PWMCL00	RTPOE00	RTPM00n	RTBH00m/ RTBL00m	Pin P4n Status
1	×	×	×	×	×	×	×	Input port
0	1	×	×	×	×	×	×	“high” output
	0	0	×	×	0	×	×	“low” output
					1	0	×	“low” output
						1	0	“low” output
							1	“high” output
		1	0	0	0	×	×	“low” output
					1	0	×	“low” output
						1	0	“low” output
							1	“high” output
				1	0	×	×	“TO50” output
					1	0	×	“TO50” output
						1	0	“TO50” output
							1	“high” output
			1	0	0	×	×	“high” output
					1	0	×	“high” output
						1	0	“high” output
							1	“low” output
				1	0	×	×	“TO50” output
					1	0	×	“TO50” output
						1	0	“TO50” output
							0	“low” output

PM4n: it n of port mode register 4 (PM4)

P4n: it n of port 4 (P4)

DCEN00: it 7 of DC control register 00 (DCCTL00)

INV00: it 4 of DCCTL00

PWMCH00: it 6 of DCCTL00

PWMCL00: it 5 of DCCTL00

RTPOE00: it 7 of real-time output port control register 0 (RTPC00)

RTPM00n: it n of real-time output port mode register 0 (RTPM00)

RTBH00m: it m of real-time output buffer register 0H (RTBH00)

RTBL00m: it m of real-time output buffer register 0L (RTBL00)

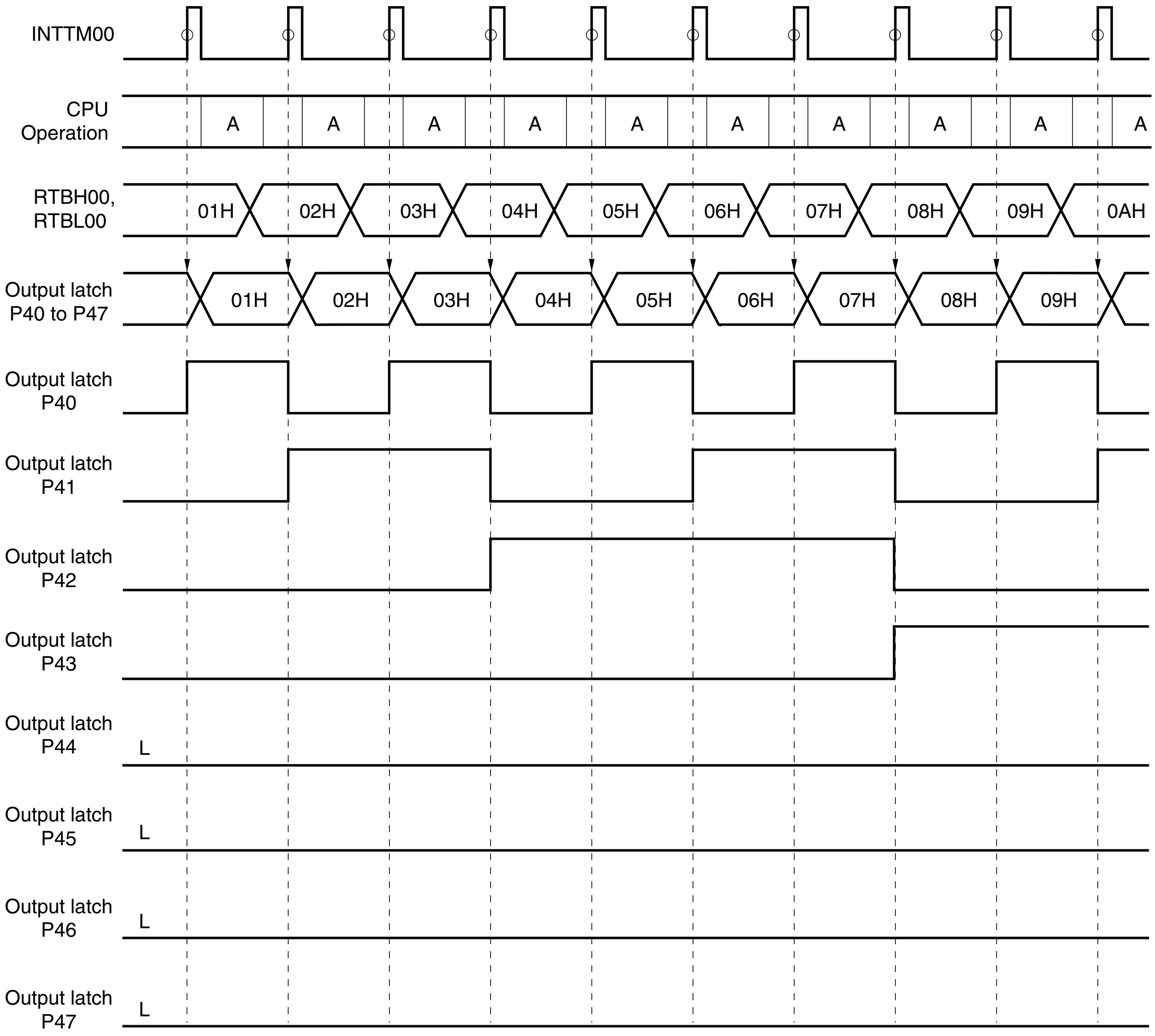
n = 0 to 7

m = 0 to 3

×: don’t care.

Figure 13-11. Real-Time Output Port Operation Timing Example (8 Bits × 1) (1/3)

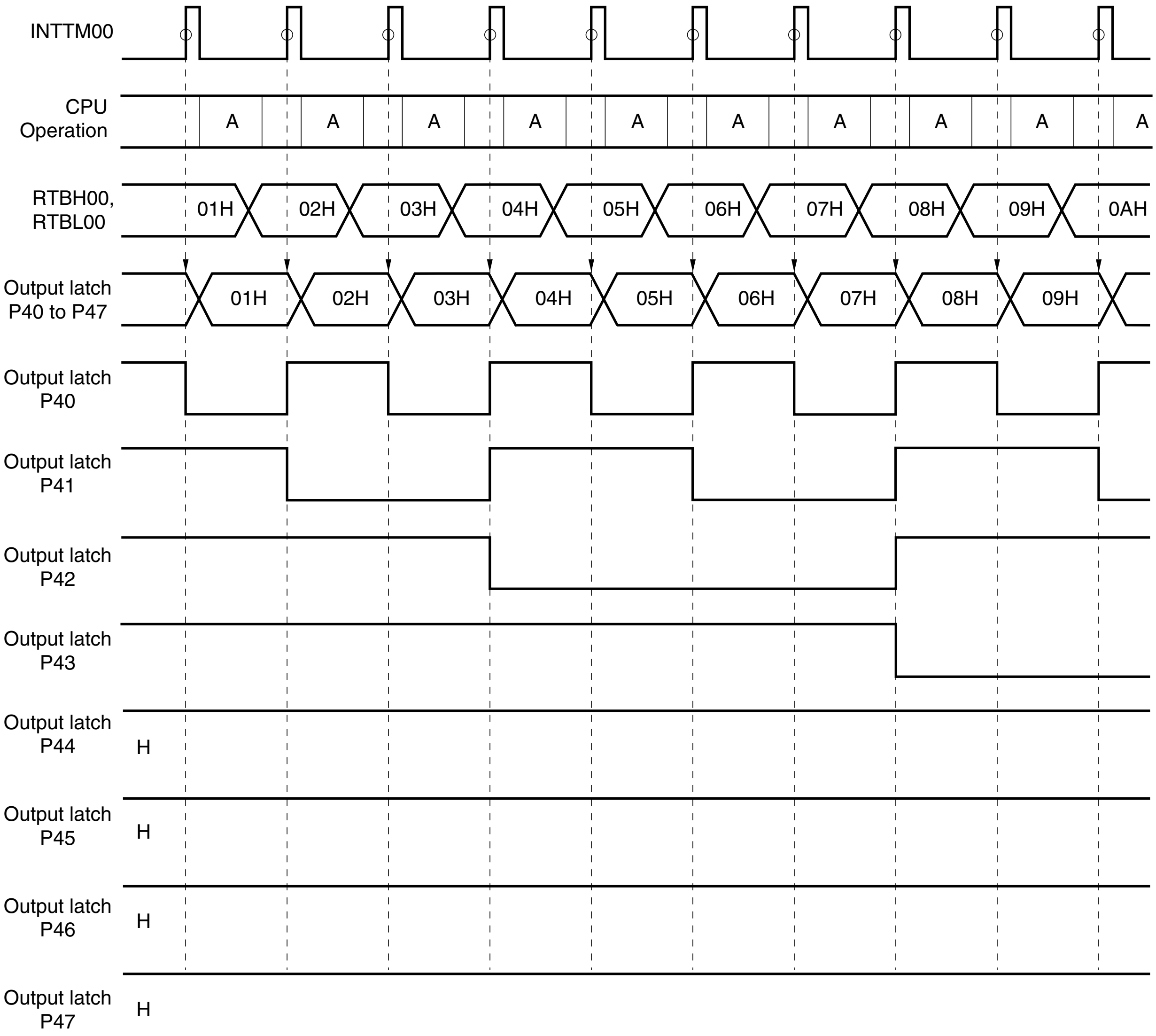
(a) 8 bits × 1 channel, inverted output disabled, no PWM modulation (EXTR00 = 0, BYTE00 = 1, INV00 = 0, PWMCH00 = 0, PWMCL00 = 0)



A: INTTM00 software processing (RTBH00, RTBL00 write)

Figure 13-11. Real-Time Output Port Operation Timing Example (8 Bits × 1) (2/3)

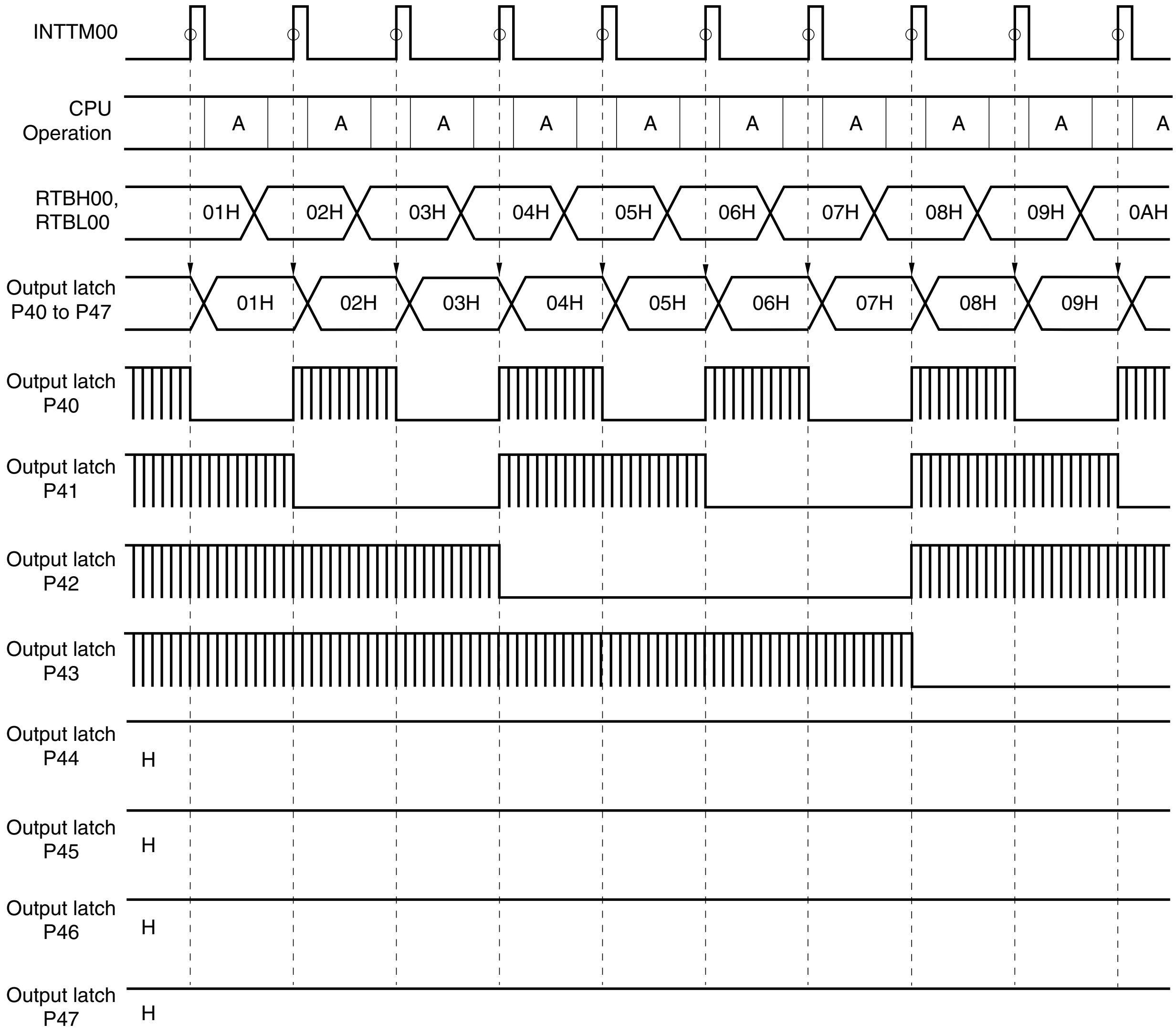
(b) 8 bits × 1 channel, inverted output enabled, no PWM modulation
(EXTR00 = 0, BYTE00 = 1, INV00 = 1, PWMCH00 = 0, PWMCL00 = 0)



A: INTTM00 software processing (RTBH00, RTBL00 write)

Figure 13-11. Real-Time Output Port Operation Timing Example (8 Bits × 1) (3/3)

(c) 8 bits × 1 channel, inverted output enabled, PWM modulation (EXTR00 = 0, BYTE00 = 1, INV00 = 1, PWMCH00 = 1, PWMCL00 = 1)



A: INTTM00 software processing (RTBH00, RTBL00 write)

(2) Using RTP10 to RTP15 as a real-time output port ..... Real-time output port 1 (6 bits × 1, or 4 bits × 1)

If real-time output is enabled when bit 7 (RTPOE01) of real-time output port control register 1 (RTPC01) is 1, the data of real-time output buffer register 1 (RTBH01, RTBL01) is transferred to the output latch in synchronization with the generation of INTTM01. Of the transferred data, only the data of the bit specified as the real-time output port by setting real-time output port mode register 1 (RTPM01) is output from bits RTP10 to RTP15. It is possible to use RTP10 to RTP15 as inverter timer output when inverter timer output is specified by DCEN01.

The operation mode can be selected as 6 bits × 1, or 4 bits × 1, by setting BYTE01.

By setting INV01, it is possible to invert the output waveform. Also, by setting PWMCL01 and PWMCH01, it is possible to perform PWM modulation of the output pattern.

If real-time output was disabled (RTPOE01 = 0) when RTPM01n = 1 and INV01 = 0, then RTP10 to RTP15 output 0.

The relationship between the settings for each bit of the control register and the real-time output is shown in Table 13-7, and an example of the operation timing is shown in Figure 13-12.

Remark

BYTE01:	Bit 5 of real-time output port control register 1 (RTPC01)
DCEN01:	Bit 7 of DC control register 1 (DCCTL1)
INV01:	Bit 4 of DC control register 1 (DCCTL1)
PWMCL01:	Bit 5 of DC control register 1 (DCCTL1)
PWMCH01:	Bit 6 of DC control register 1 (DCCTL1)
RTPM01n:	Bit n (n = 0 to 5) of real-time output port mode register 1 (RTPM01).

<R>

Table 13-7. Relationship Between Settings of Each Bit of Control Register and Real-Time Output

CE0	DCEN01	INV01	PWMCH01/ PWMCL01	RTPOE01	RTPM01n	RTBH01m/ RTBL01m	Pin TW0TOn Status
0	×	×	×	×	×	×	Hi-Z
1	0	×	×	×	×	×	TW0TOn
	1	0	0	0	×	×	“low” output
				1	0	×	“low” output
					1	0	“low” output
						1	“high” output
			1	0	×	×	TW0TO0
				1	0	×	TW0TO0
					1	0	TW0TO0
						1	“high” output
		1	0	0	×	×	“high” output
				1	0	×	“high” output
					1	0	“high” output
						1	“low” output
			1	0	×	×	$\overline{\text{TW0TO0}}$
				1	0	×	$\overline{\text{TW0TO0}}$
					1	0	$\overline{\text{TW0TO0}}$
						1	“low” output

CE0: Bit 7 of inverter timer control register (TW0C)

DCEN01: Bit 7 of DC control register 01 (DCCTL01)

INV01: Bit 4 of DCCTL01

PWMCH01: Bit 6 of DCCTL01

PWMCL01: Bit 5 of DCCTL01

RTPOE01: Bit 7 of real-time output port control register 1 (RTPC01)

RTPM01n: Bit n of real-time output port mode register 1 (RTPM01)

RTBH01m: Bit m of real-time output buffer register 1H (RTBH01)

RTBL01m: Bit m of real-time output buffer register 1L (RTBL01)

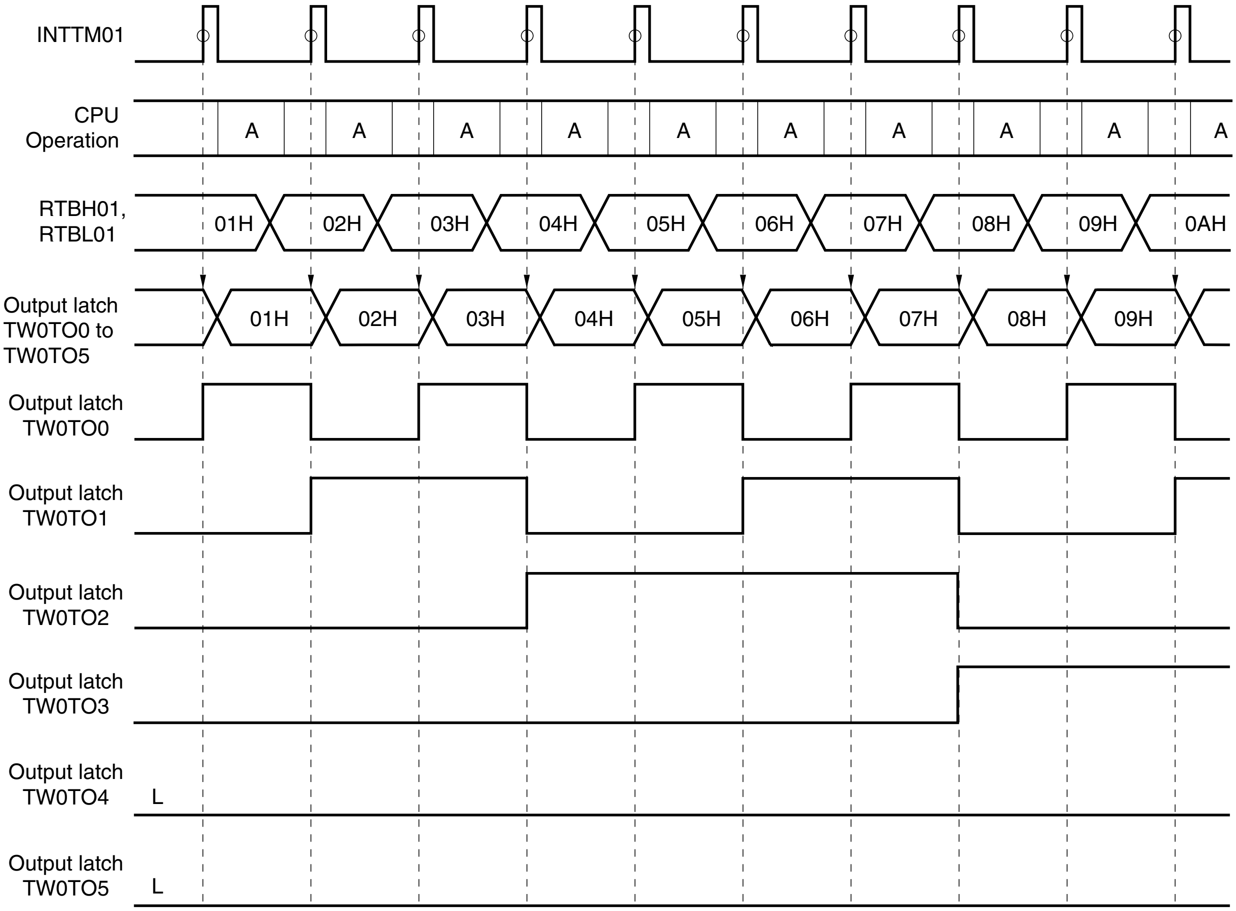
n = 0 to 5

m = 0 to 3

×: don’t care.

Figure 13-12. Real-Time Output Port Operation Timing Example (6 Bits × 1) (1/3)

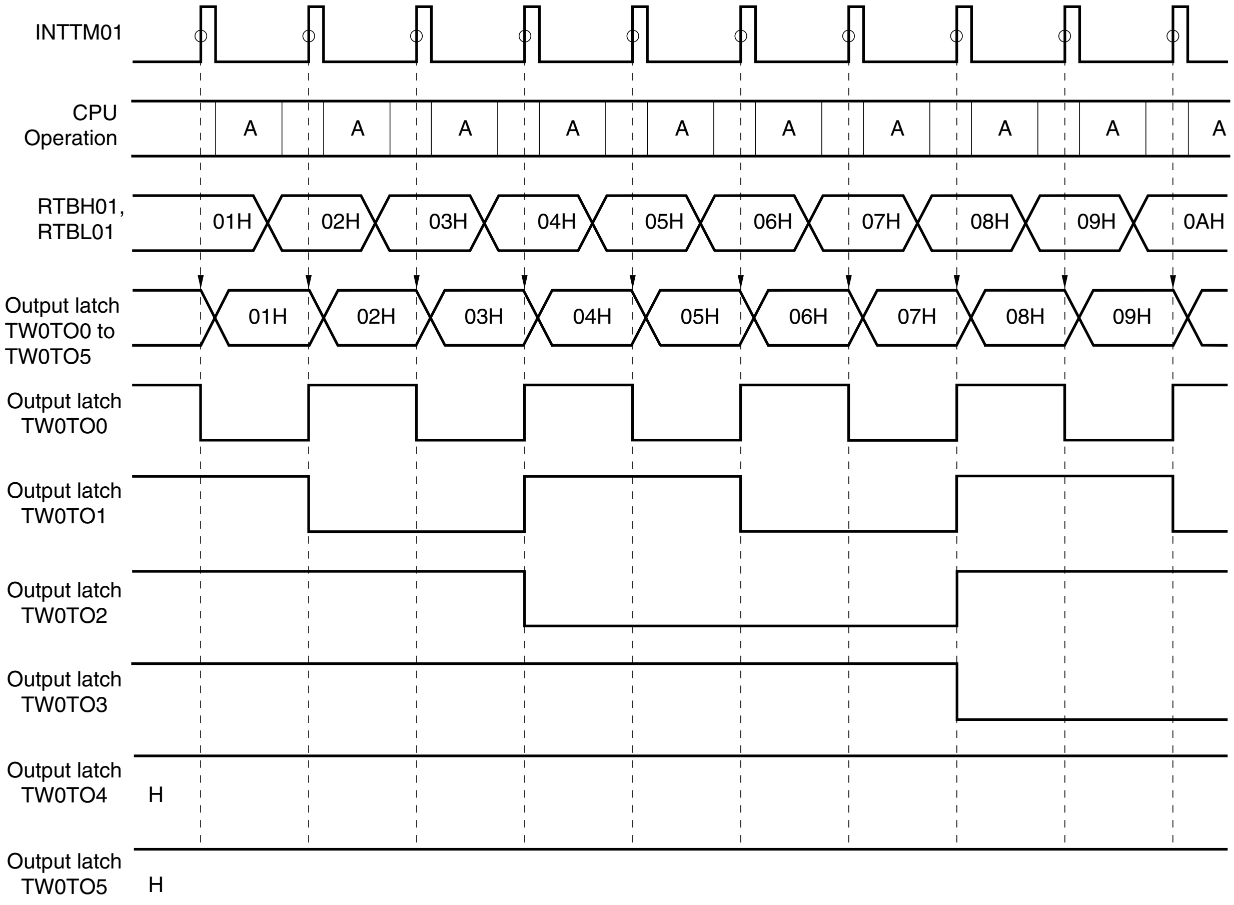
(a) 6 bits × 1 channel, inverted output disabled, no PWM modulation (BYTE01 = 1, INV01 = 0, PWMCH01 = 0, PWMCL01 = 0)



A: INTTM01 software processing (RTBH01, RTBL01 write)

Figure 13-12. Real-Time Output Port Operation Timing Example (6 Bits × 1) (2/3)

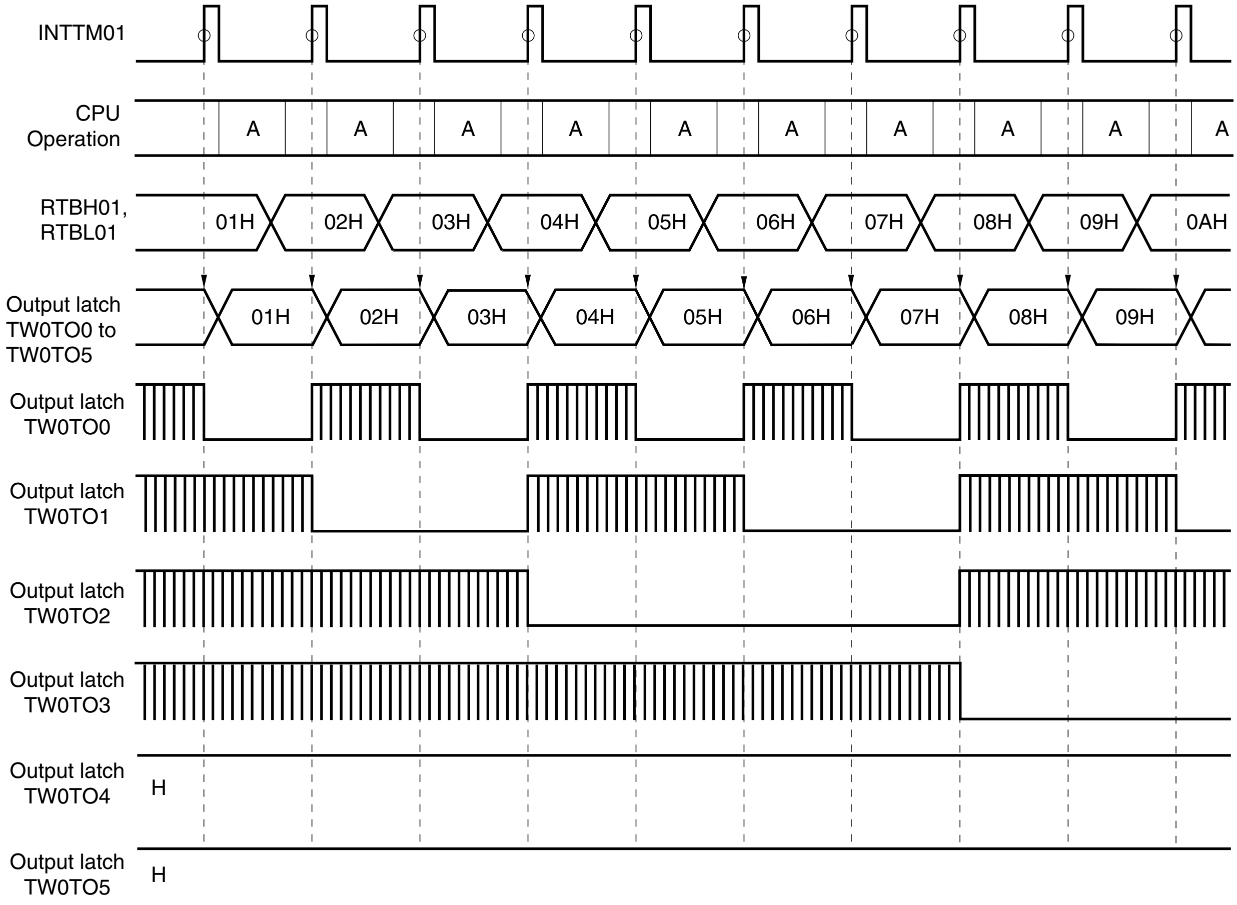
(b) 6 bits × 1 channel, inverted output enabled, no PWM modulation (BYTE01 = 1, INV01 = 1, PWMCH01 = 0, PWMCL01 = 0)



A: INTTM01 software processing (RTBH01, RTBL01 write)

Figure 13-12. Real-Time Output Port Operation Timing Example (6 Bits × 1) (3/3)

(c) 6 bits × 1 channel, inverted output enabled, PWM modulation (BYTE01 = 1, INV01 = 1, PWMCH01 = 1, PWMCL01 = 1)



A: INTTM01 software processing (RTBH01, RTBL01 write)

13.5 Using Real-Time Output Port

When using the real-time output port, perform the following steps.

(1) Disable real-time output operation.
Clear bit 7 (RTPOE0n) of real-time output port control register n (RTPC0n) to 0.

(2) Initial setting

- Set the initial value to the port output latch (real-time output port 0 only).
- Specify the real-time output port mode in 1-bit units.
  Set real-time output port mode register n (RTPM0n).
- Select the operation mode (trigger and a valid edge).
  Set bits 4, 5, and 6 (EXTR00, BYTE00, and RTPEG00) of RTPC00 or set bit 5 (BYTE01) of RTPC01.
- For real-time output port 0, set an initial value equal to the port output latch in real-time output buffer register
  0 (RTBH00, RTBL00).
  For real-time output port 1, set an initial value in real-time output buffer register 1 (RTBH01, RTBL01).
- Set DC control register 0n (DCCTL0n).

(3) Enable the real-time output operation.
RTPOE0n = 1

(4) Set the port output latch to 0 (only for real-time output port 0).

Remark For real-time output port 0, the value output by the real-time output operation is the ORed value of the output latch of the port and real-time output (see Figure 13-1 (a)). Therefore, when real-time output port 0 is used, the port output latch should be set to 0 after the real-time output operation is enabled (RTPOE00 = 0 → 1) until the first transfer trigger is generated.

(5) Set the next output to RTBH0n and RTBL0n before the selected transfer trigger is generated.

(6) Sequentially set the next real-time output value to RTBH0n and RTBL0n by using the interrupt servicing corresponding to the selected trigger.

Remark n = 0, 1

13.6 Notes on Real-Time Output Port

(1) Before performing the initial setting, disable the real-time output operation by clearing bit 7 (RTPOE0n) of real-time output port control register n (RTPC0n) to 0 (n = 0, 1).

(2) Once the real-time output operation has been disabled (RTPOE0n = 0), be sure to set the same initial value as the output latch to real-time output buffer register n (RTBH0n and RTBL0n) before enabling the real-time output operation (RTPOE0n = 0 → 1) (n = 0, 1).

CHAPTER 14 DC INVERTER CONTROL FUNCTION

The μ PD78F0714 realizes a 3-phase PWM DC inverter control by combination of 10-bit inverter control timer and real-time output port.

See the following chapters.

- CHAPTER 6 10-BIT INVERTER CONTROL TIMER
- CHAPTER13 REAL-TIME OUTPUT PORT

<R> Refer to the following application notes for application systems that use DC inverter control.

- Motor Control by μPD78F0714 Sensorless (BEMF) 120° Excitation Method (U18051E)
- Motor Control by μPD78F0714 Hall IC 120° Excitation Method (U18774E)

15.1 Functions of A/D Converter

The A/D converter converts an analog input signal into a digital value, and consists of up to eight channels (ANI0 to ANI7) with a resolution of 10 bits.

The A/D converter has the following two functions.

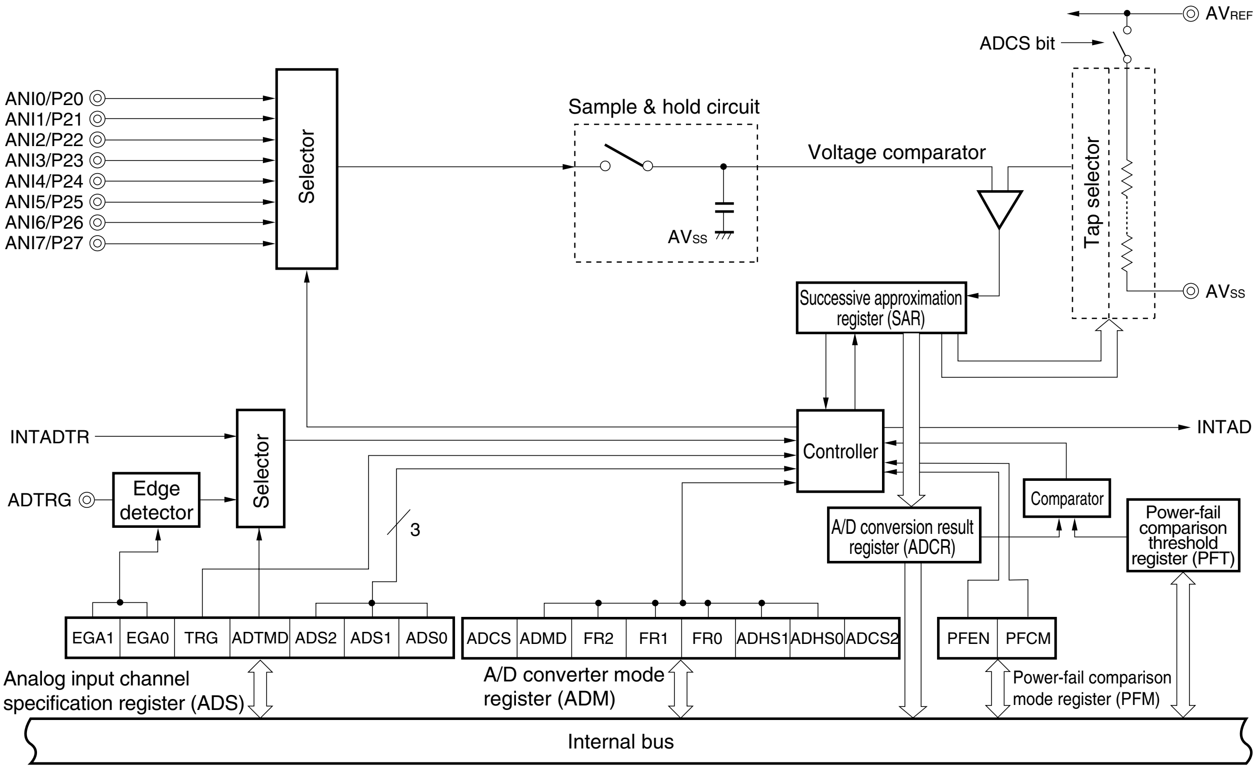
(1) 10-bit resolution A/D conversion

10-bit resolution A/D conversion is carried out repeatedly for one channel selected from analog inputs ANI0 to ANI7. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated.

(2) Power-fail detection function

This function is used to detect a voltage drop in a battery. The A/D conversion result (ADCR register value) and power-fail comparison threshold register (PFT) value are compared. INTAD is generated only when a comparative condition has been matched.

Figure 15-1. Block Diagram of A/D Converter



15.2 Configuration of A/D Converter

The A/D converter consists of the following hardware.

Table 15-1. Registers of A/D Converter Used on Software

Item	Configuration
Registers	Successive approximation register (SAR) A/D conversion result register (ADCR) A/D converter mode register (ADM) Analog input channel specification register (ADS) Power-fail comparison mode register (PFM) Power-fail comparison threshold register (PFT)

(1) ANI0 to ANI7 pins

These are the analog input pins of the 8-channel A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin by the analog input channel specification register (ADS) can be used as input port pins.

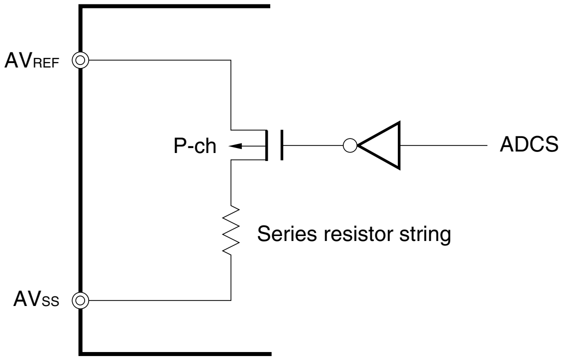
(2) Sample & hold circuit

The sample & hold circuit samples the input signal of the analog input pin selected by the selector when A/D conversion is started, and holds the sampled analog input voltage value during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AV_{REF} and AV_{SS}, and generates a voltage to be compared with the analog input signal.

Figure 15-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled analog input voltage and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register compares the sampled analog voltage and the voltage of the series resistor string, and converts the result, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) A/D conversion result register (ADCR)

The result of A/D conversion is loaded from the successive approximation register (SAR) to this register each time A/D conversion is completed, and the ADCR register holds the result of A/D conversion in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) Controller

When A/D conversion has been completed or when the power-fail detection function is used, this controller compares the result of A/D conversion (value of the ADCR register) and the value of the power-fail comparison threshold register (PFT). It generates the interrupt INTAD only if a specified comparison condition is satisfied as a result.

(8) AV_{REF} pin

This pin inputs an analog power/reference voltage to the A/D converter. Always use this pin at the same potential as that of the V_{DD} pin even when the A/D converter is not used.

The signal input to ANI0 to ANI7 is converted into a digital signal, based on the voltage applied across AV_{REF} and AV_{SS}.

(9) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

(10) A/D converter mode register (ADM)

This register is used to set the conversion time of the analog input signal to be converted, and to start or stop the conversion operation.

(11) Analog input channel specification register (ADS)

This register is used to specify the port that inputs the analog voltage to be converted into a digital signal.

(12) Power-fail comparison mode register (PFM)

This register is used to set the power-fail monitor mode.

(13) Power-fail comparison threshold register (PFT)

This register is used to set the threshold value that is to be compared with the value of the A/D conversion result register (ADCR).

15.3 Registers Used in A/D Converter

The A/D converter uses the following five registers.

- A/D converter mode register (ADM)
- Analog input channel specification register (ADS)
- A/D conversion result register (ADCR)
- Power-fail comparison mode register (PFM)
- Power-fail comparison threshold register (PFT)

(1) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

ADM can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 15-3. Format of A/D Converter Mode Register (ADM)

Address: FF6CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM	ADCS	ADMD	FR2[Note 1]	FR1[Note 1]	FR0[Note 1]	ADHS1[Note 1]	ADHS0[Note 1]	ADCS2

ADCS	A/D conversion operation control
0	Stops conversion operation
1	Enables conversion operation

ADMD	Operation mode control
0	Select mode
1	Scan mode

<R>

FR2	FR1	FR0	ADHS1	ADHS0	A/D conversion time selection
×	×	×	0	0	Setting prohibited
0	0	0	0	1	96/f_X
0	0	1	0	1	72/f_X
0	1	0	0	1	48/f_X
0	1	1	0	1	24/f_X
1	0	0	0	1	224/f_X
1	0	1	0	1	168/f_X
1	1	0	0	1	112/f_X
1	1	1	0	1	56/f_X
0	0	0	1	0	72/f_X
0	0	1	1	0	54/f_X
0	1	0	1	0	36/f_X
0	1	1	1	0	18/f_X
1	×	×	1	0	Setting prohibited
×	×	×	1	1	Setting prohibited

ADCS2	Boost reference voltage generator operation control[Note 2]
0	Stops operation of reference voltage generator
1	Enables operation of reference voltage generator

Notes 1. Select the A/D conversion time in the combination of FR2 to FR0, ADHS1, and ADHS0. For details of A/D conversion time, see **Table 15-3.**

2. A booster circuit is incorporated to realize low-voltage operation. The operation of the circuit that generates the reference voltage for boosting is controlled by ADCS2, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCS2 is set to 1, the conversion result at that time has priority over the first conversion result.

Remark f_X: X1 input clock oscillation frequency

(a) Controlling reference voltage generator for boosting

When the ADCS2 bit = 0, power to the A/D converter drops. The converter requires a setup time of 1 μs or more after the ADCS2 bit has been set to 1.

Therefore, the result of A/D conversion becomes valid from the first result by setting the ADCS bit to 1 at least 1 μs after the ADCS2 bit has been set to 1.

Table 15-2. Settings of ADCS and ADCS2

ADCS	ADCS2	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (only reference voltage generator consumes power)
1	0	Conversion mode (reference voltage generator operation stopped[Note 1])
1	1	Conversion mode (reference voltage generator operates[Note 2])

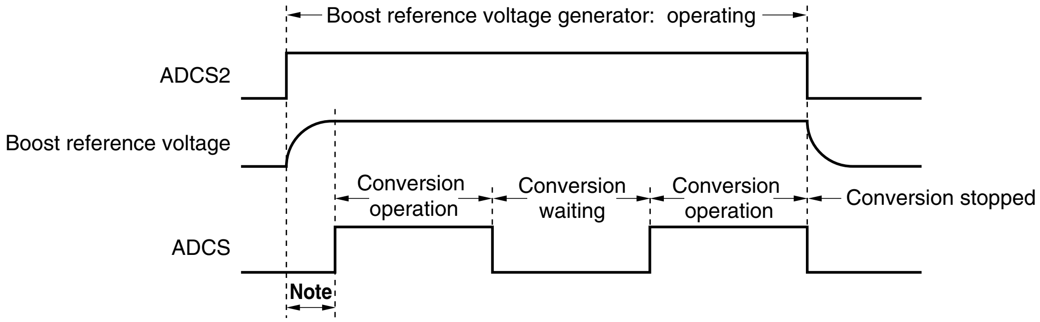
Notes 1. If the ADCS and ADCS2 bits are changed from 00B to 10B, the reference voltage generator for boosting automatically turns on. If the ADCS bit is cleared to 0 while the ADCS2 bit is 0, the voltage generator automatically turns off. In the software trigger mode (ADS.TRG bit = 0), use of the first A/D conversion result is prohibited.

In the hardware trigger mode (TRG bit = 1), use the A/D conversion result only if A/D conversion is started after the lapse of the oscillation stabilization time of the reference voltage generator for boosting.

2. If the ADCS and ADCS2 bits are changed from 00B to 11B, the reference voltage generator for boosting automatically turns on. If the ADCS bit is cleared to 0 while the ADCS2 bit is 1, the voltage generator stays on. In the software trigger mode (TRG bit = 0), use of the first A/D conversion result is prohibited.

In the hardware trigger mode (TRG bit = 1), use the A/D conversion result only if A/D conversion is started after the lapse of the oscillation stabilization time of the reference voltage generator for boosting.

Figure 15-4. Timing Chart When Boost Reference Voltage Generator Is Used



Note The time from the rising of the ADCS2 bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the reference voltage.

Cautions 1. A/D conversion must be stopped before rewriting bits FR0 to FR2, ADHS0, and ADHS1 to values other than the identical data.

2. If data is written to ADM, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

<R>

Table 15-3. A/D Conversion Time

FR2	FR1	FR0	ADHS1	ADHS0	Conversion Time (t_{CONV})					
						f_X = 20 MHz	f_X = 16 MHz	f_X = 10 MHz	f_X = 8.38 MHz	f_X = 5 MHz
×	×	×	0	0	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	0	0	0	1	96/f_X	4.8 µs	6 µs	9.6 µs	11.5 µs	19.2 µs
0	0	1	0	1	72/f_X	3.6 µs[Note]	4.5 µs[Note]	7.2 µs	8.6 µs	14.4 µs
0	1	0	0	1	48/f_X	Setting prohibited	Setting prohibited	4.8 µs	5.8 µs	9.6 µs
0	1	1	0	1	24/f_X	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	4.8 µs
1	0	0	0	1	224/f_X	11.2 µs	14 µs	22.4 µs	26.8 µs	44.8 µs
1	0	1	0	1	168/f_X	8.4 µs	10.5 µs	16.8 µs	20.1 µs	33.6 µs
1	1	0	0	1	112/f_X	5.6 µs	7 µs	11.2 µs	13.4 µs	22.4 µs
1	1	1	0	0	56/f_X	Setting prohibited	4.5 µs[Note]	5.6 µs	6.7 µs	11.2 µs
0	0	0	1	0	72/f_X	3.6 µs[Note]	Setting prohibited	7.2 µs	8.6 µs	14.4 µs
0	0	1	1	0	54/f_X	Setting prohibited	Setting prohibited	5.4 µs	6.5 µs	10.8 µs
0	1	0	1	0	36/f_X	Setting prohibited	Setting prohibited	3.6 µs[Note]	4.3 µs[Note]	7.2 µs
0	1	1	1	0	18/f_X	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	3.6 µs[Note]
1	×	×	1	0	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
×	×	×	1	1	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited

Note If 3.6 µs ≤ t_{CONV} < 4.8 µs, this can be set only at AV_{REF} ≥ 4.5 V.

Remark f_X: X1 input clock oscillation frequency

(2) Analog input channel specification register (ADS)

This register specifies the input port of the analog voltage to be A/D converted.

ADS can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 15-5. Format of Analog Input Channel Specification Register (ADS)

Address: FF6DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	EGA1	EGA0	TRG	ADTMD	0	ADS2	ADS1	ADS0

EGA1[Note 1]	EGA0[Note 1]	Specification of external trigger signal (ADTRG) edge
0	0	No edge detection
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

TRG	Trigger mode selection
0	Software trigger mode
1	Hardware trigger mode

ADTMD[Note 2]	Specification of hardware trigger mode
0	External trigger (ADTRG pin input)
1	Timer trigger (INTADTR signal generated)

ADS2	ADS1	ADS0	Analog input channel specification	
			Select mode	Scan mode
0	0	0	ANI0	ANI0
0	0	1	ANI1	ANI0, ANI1
0	1	0	ANI2	ANI0 to ANI2
0	1	1	ANI3	ANI0 to ANI3
1	0	0	ANI4	ANI0 to ANI4
1	0	1	ANI5	ANI0 to ANI5
1	1	0	ANI6	ANI0 to ANI6
1	1	1	ANI7	ANI0 to ANI7

Notes 1. The EGA1 and EGA0 bits are valid only when the hardware trigger mode (TRG bit = 1) and external trigger mode (ADTRG pin input: ADTMD bit = 1) are selected.

2. The ADTMD bit is valid only when the hardware trigger mode (TRG bit = 1) is selected.

Cautions 1. Be sure to clear bit 3 of ADS to 0.

2. If data is written to ADS, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

(3) A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register.

The lower 6 bits are fixed to 0. The conversion result bits are stored in ADCR in order from the MSB. The higher 8 bits of the conversion result are stored in FF1BH and the lower 2 bits in FF1AH.

ADCR can be read by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes ADCR undefined.

Figure 15-6. Format of A/D Conversion Result Register (ADCR)

Address: FF1AH, FF1BH After reset: Undefined R

Symbol	FF1BH								FF1AH							
ADCR											0	0	0	0	0	0

Cautions 1. When writing to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using timing other than the above may cause an incorrect conversion result to be read.

2. If data is read from ADCR, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

(4) Power-fail comparison mode register (PFM)

The power-fail comparison mode register (PFM) is used to compare the A/D conversion result (value of the ADCR register) and the value of the power-fail comparison threshold register (PFT).

PFM can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 15-7. Format of Power-Fail Comparison Mode Register (PFM)

Address: FF6EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFM	PFEN	PFCM	0	0	0	0	0	0

PFEN	Power-fail comparison enable
0	Stops power-fail comparison (used as a normal A/D converter)
1	Enables power-fail comparison (used for power-fail detection)

PFCM		Power-fail comparison mode selection
0	Higher 8 bits of ADCR ³ PFT	Interrupt request signal (INTAD) generation
	Higher 8 bits of ADCR < PFT	No INTAD generation
1	Higher 8 bits of ADCR ³ PFT	No INTAD generation
	Higher 8 bits of ADCR < PFT	INTAD generation

Caution If data is written to PFM, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

(5) Power-fail comparison threshold register (PFT)

The power-fail comparison threshold register (PFT) is a register that sets the threshold value when comparing the values with the A/D conversion result.

8-bit data in PFT is compared to the higher 8 bits (FF1BH) of the 10-bit A/D conversion result.

PFT can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 15-8. Format of Power-Fail Comparison Threshold Register (PFT)

Address: FF6FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFT	PFT7	PFT6	PFT5	PFT4	PFT3	PFT2	PFT1	PFT0

Caution If data is written to PFT, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

15.4 Relationship Between Input Voltage and A/D Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI7) and the theoretical A/D conversion result (stored in the A/D conversion result register (ADCR)) is shown by the following expression.

$$\text{SAR} = \text{INT}\left(\frac{V_{\text{AIN}}}{AV_{\text{REF}}} \times 1024 + 0.5\right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

or

$$(\text{ADCR} - 0.5) \times \frac{AV_{\text{REF}}}{1024} \leq V_{\text{AIN}} < (\text{ADCR} + 0.5) \times \frac{AV_{\text{REF}}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

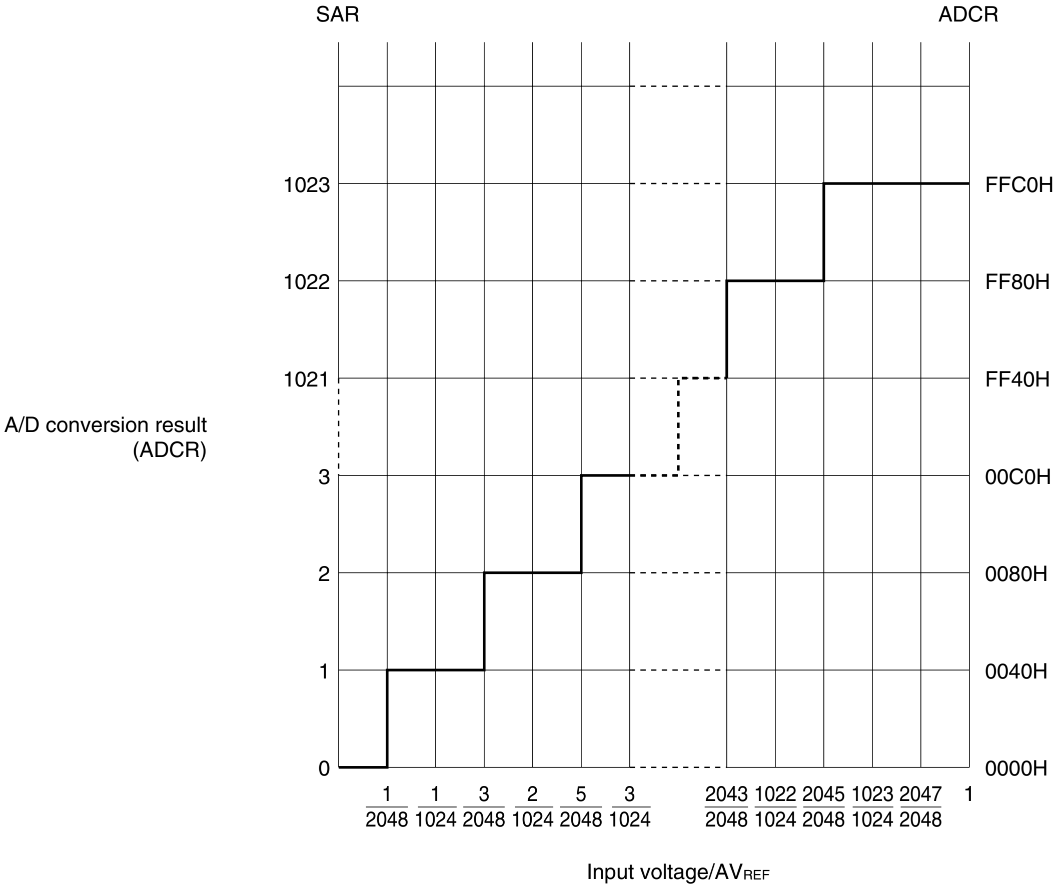
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 15-9 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 15-9. Relationship Between Analog Input Voltage and A/D Conversion Result



15.5 A/D Converter Operations

15.5.1 Basic operations of A/D converter

<1> Select one channel for A/D conversion using the analog input channel specification register (ADS).
Select the conversion time by using the FR2 to FR0, ADHS1, and ADSH0 bits of the A/D converter mode register (ADM).

<2> Set ADCS2 to 1 and wait for 1 μs or longer.

<3> Set ADCS to 1 and start the conversion operation.
(<4> to <10> are operations performed by hardware.)

<4> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<5> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the input analog voltage is held until the A/D conversion operation has ended.

<6> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} by the tap selector.

<7> The voltage difference between the series resistor string voltage tap and analog input is compared by the voltage comparator. If the analog input is greater than (1/2) AV_{REF}, the MSB of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.

<8> Next, bit 8 of SAR is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and analog input voltage are compared and bit 8 of SAR is manipulated as follows.

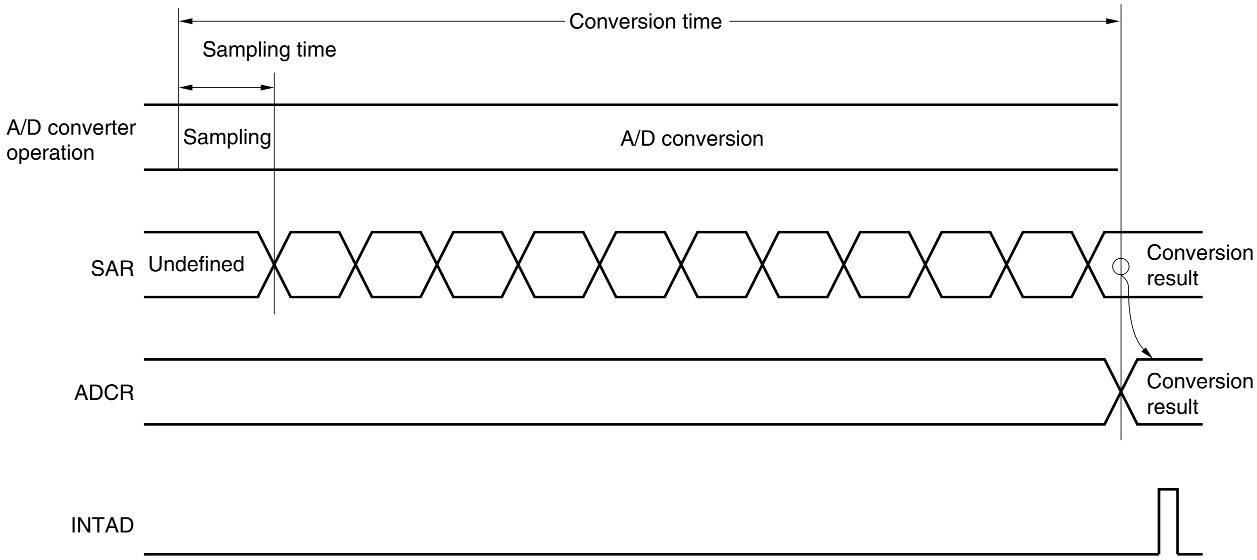
- Analog input voltage ≥ Voltage tap: Bit 8 = 1
- Analog input voltage < Voltage tap: Bit 8 = 0

<9> Comparison is continued in this way up to bit 0 of SAR.

<10> Upon completion of the comparison of 10 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR) and then latched.
At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<11> Repeat steps <4> to <10>, until ADCS is cleared to 0.
To stop the A/D converter, clear ADCS to 0.
To restart A/D conversion from the status of ADCS2 = 1, start from <3>. To restart A/D conversion from the status of ADCS2 = 0, however, start from <2>.

Figure 15-10. Basic Operation of A/D Converter



A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to one of the ADM, analog input channel specification register (ADS), power-fail comparison mode register (PFM), or power-fail comparison threshold register (PFT) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

$\overline{\text{RESET}}$ input makes the A/D conversion result register (ADCR) undefined.

15.5.2 Trigger modes

The μPD78F0714 has the following three trigger modes that set the A/D conversion start timing. These trigger modes are set by the ADS register.

- Software trigger mode
- External trigger mode (hardware trigger mode)
- Timer trigger mode (hardware trigger mode)

(1) Software trigger mode

This mode is used to start A/D conversion by setting the ADM.ADCS bit to 1 while the ADS.TRG bit is 0.

Conversion is repeatedly performed as long as the ADCS bit is not cleared to 0 after completion of A/D conversion.

If the ADM, ADS, PFM, or PFT register is written during conversion, A/D conversion is aborted and started again from the beginning.

(2) External trigger mode (hardware trigger mode)

This is the status in which the ADS.TRG bit is set to 1 and ADS.ADTMD bit is cleared to 0. This mode is used to start A/D conversion by detecting an external trigger (ADTRG) after the ADCS bit has been set to 1.

The A/D converter waits for the external trigger (ADTRG) after the ADCS bit is set to 1.

The valid edge of the signal input to the ADTRG pin is specified by using the ADS.EGA1 and ADS.EGA0 bits. When the specified valid edge is detected, A/D conversion is started.

When A/D conversion is completed, the A/D converter waits for the external trigger (ADTRG) again.

If a valid edge is input to the ADTRG pin during A/D conversion, A/D conversion continues without detecting the trigger.

If the ADM, ADS, PFM, or PFT register is written during conversion, A/D conversion is aborted and the A/D converter waits for an external trigger (ADTRG).

(3) Timer trigger mode (hardware trigger mode)

This mode is used to start A/D conversion by detecting a timer trigger (INTADTR) after the ADCS bit has been set to 1 with the TRG bit = 1 and ADTMD bit = 1.

The A/D converter waits for the timer trigger (INTADTR) after the ADCS bit is set to 1.

When the INTADTR signal is generated, A/D conversion is started.

When A/D conversion is completed, the A/D converter waits for the timer trigger (INTADTR) again.

If the INTADTR signal is generated during A/D conversion, A/D conversion is aborted and started again from the beginning.

If the ADM, ADS, PFM, or PFT register is written during conversion, A/D conversion is aborted and the A/D converter waits for a timer trigger (INTADTR).

15.5.3 Operation modes

The following two operation modes are available. These operation modes are set by the ADM register.

- Select mode
- Scan mode

(1) Select mode

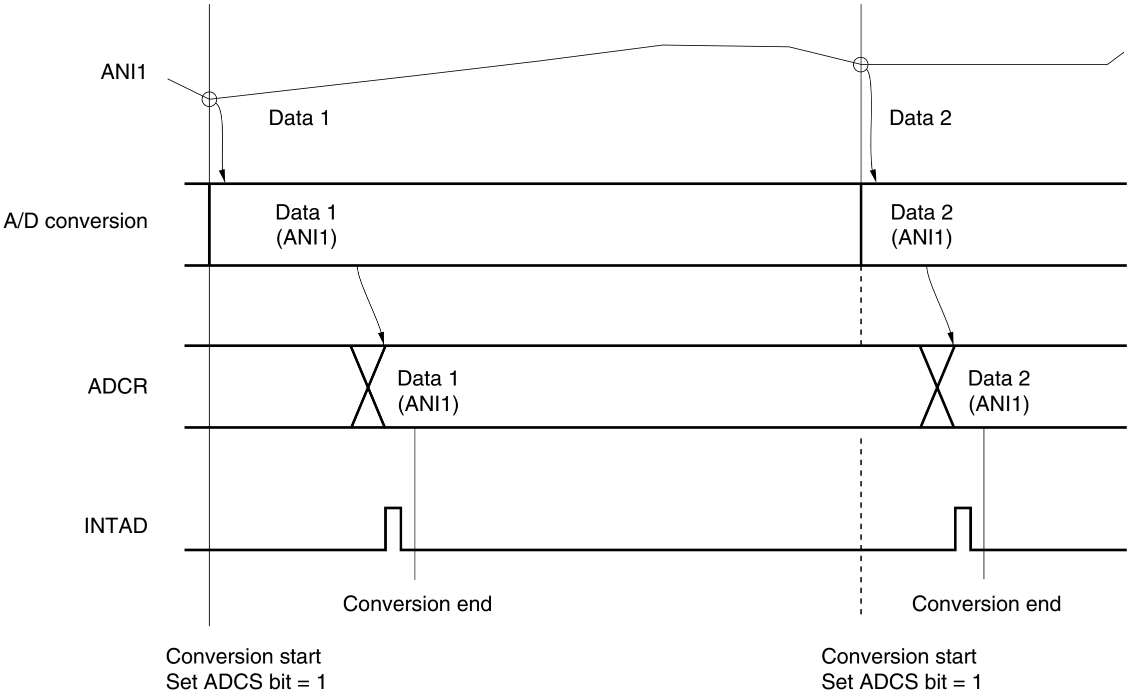
One input analog signal specified by the ADS register while the ADM.ADMD bit = 0 is converted. When conversion is complete, the result of conversion is stored in the ADCR register.

At the same time, the A/D conversion end interrupt request signal (INTAD) is generated. However, the INTAD signal may or may not be generated depending on setting of the PFM and PFT registers. For details, refer to **15.5.4 Power fail detection function**.

If anything is written to the ADM, ADS, PFM, and PFT registers during conversion, A/D conversion is aborted. In the software trigger mode, A/D conversion is started from the beginning again. In the hardware trigger mode, the A/D converter waits for a trigger.

If the trigger is detected during conversion in hardware trigger mode, A/D conversion is aborted and started again from the beginning.

Figure 15-11. Example of Select Mode Operation Timing (ADS.ADS2 to ADS.ADS0 Bits = 001B)



(2) Scan mode

In this mode, the analog signals specified by the ADS register and input from the ANI0 pin while the ADM.ADMD bit = 1 are sequentially selected and converted.

When conversion of one analog input signal is complete, the conversion result is stored in the ADCR register and, at the same time, the A/D conversion end interrupt request signal (INTAD) is generated.

The A/D conversion results of all the analog input signals are stored in the ADCR register. It is therefore recommended to save the contents of the ADCR register to RAM once A/D conversion of one analog input signal has been completed.

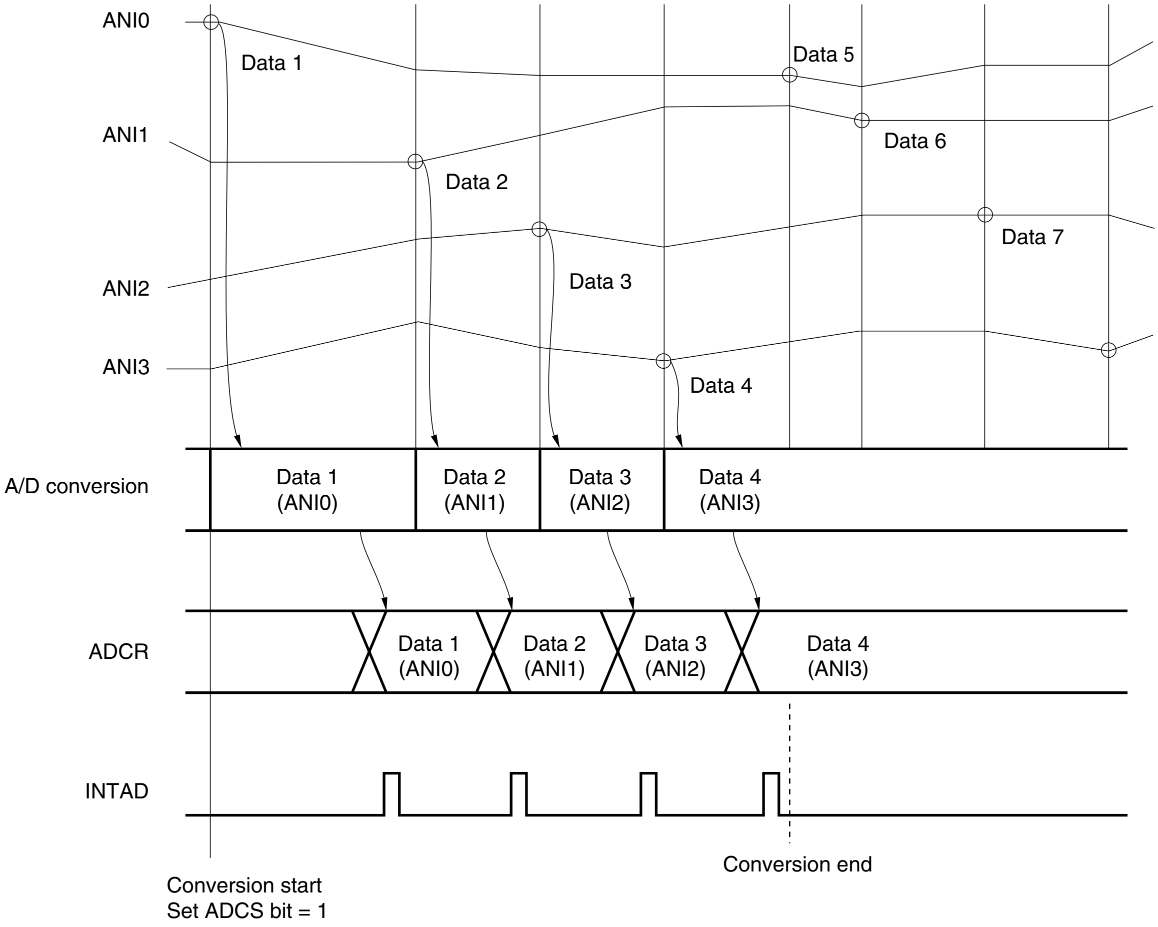
In the hardware trigger mode (ADS.TRG bit = 1), the A/D converter waits for a trigger after it has completed A/D conversion of the analog signals specified by the ADS register and input from the ANI0 pin.

If anything is written to the ADM, ADS, PFM, and PFT registers during conversion, A/D conversion is aborted. In the software trigger mode, A/D conversion is started from the beginning again. In the hardware trigger mode, the A/D converter waits for a trigger. Conversion starts again from the ANI0 pin.

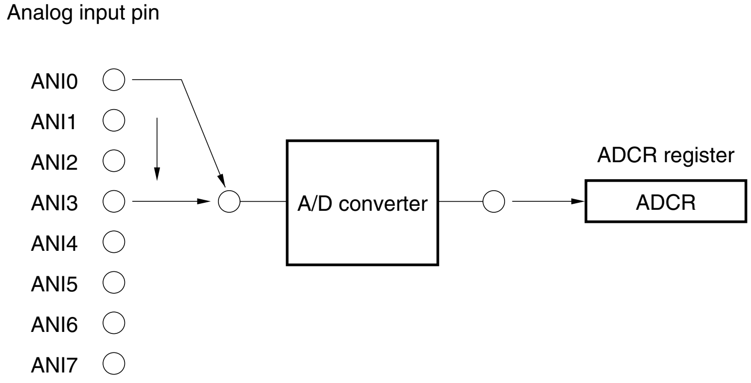
If the trigger is detected during conversion in hardware trigger mode, A/D conversion is aborted and started again from the beginning (ANI0 pin).

Figure 15-12. Example of Scan Mode Operation Timing (ADS.ADS2 to ADS.ADS0 Bits = 011B)

(a) Timing example



(b) Block diagram



15.5.4 Power-fail monitoring function

The following two functions can be selected by setting of bit 7 (PFEN) of the power-fail comparison mode register (PFM).

- Normal 10-bit A/D converter (PFEN = 0)
- Power-fail detection function (PFEN = 1)

(1) Normal A/D conversion operation (when PFEN = 0)

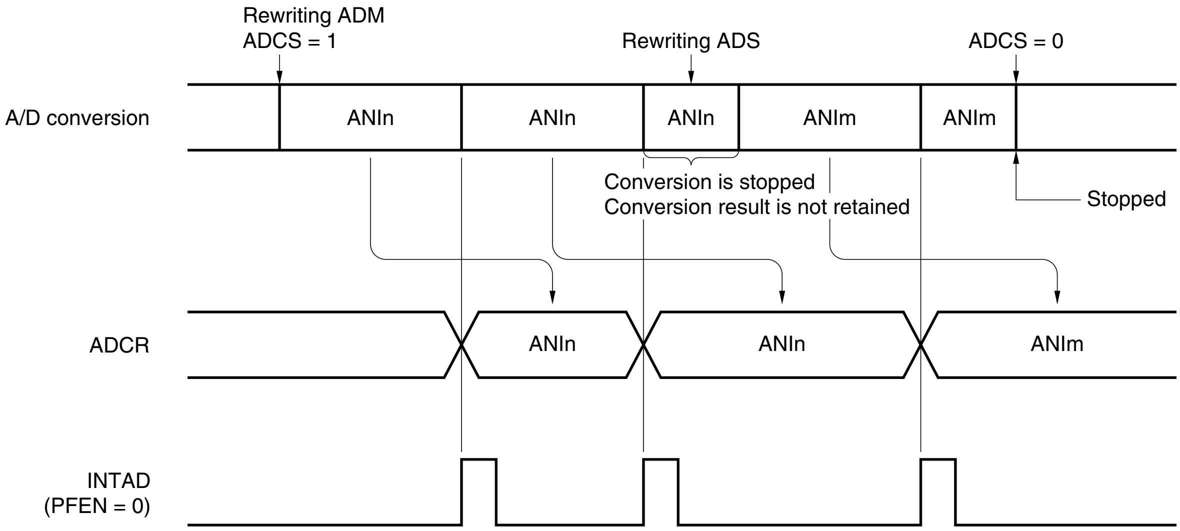
By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 0, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), and an interrupt request signal (INTAD) is generated. Once the A/D conversion has started and when one A/D conversion has been completed, the next A/D conversion operation is immediately started. The A/D conversion operations are repeated until new data is written to ADS.

If ADM, ADS, the power-fail comparison mode register (PFM), and the power-fail comparison threshold register (PFT) are rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning.

If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result is undefined.

Figure 15-13. A/D Conversion Operation



Remarks 1. n = 0 to 7

2. m = 0 to 7

(2) Power-fail detection function (when PFEN = 1)

By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1 and bit 7 (PFEN) of the power-fail comparison mode register (PFM) to 1, the A/D conversion operation of the voltage applied to the analog input pin specified by the analog input channel specification register (ADS) is started.

When the A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR), the values are compared with power-fail comparison threshold register (PFT), and an interrupt request signal (INTAD) is generated under the condition specified by bit 6 (PFCM) of PFM.

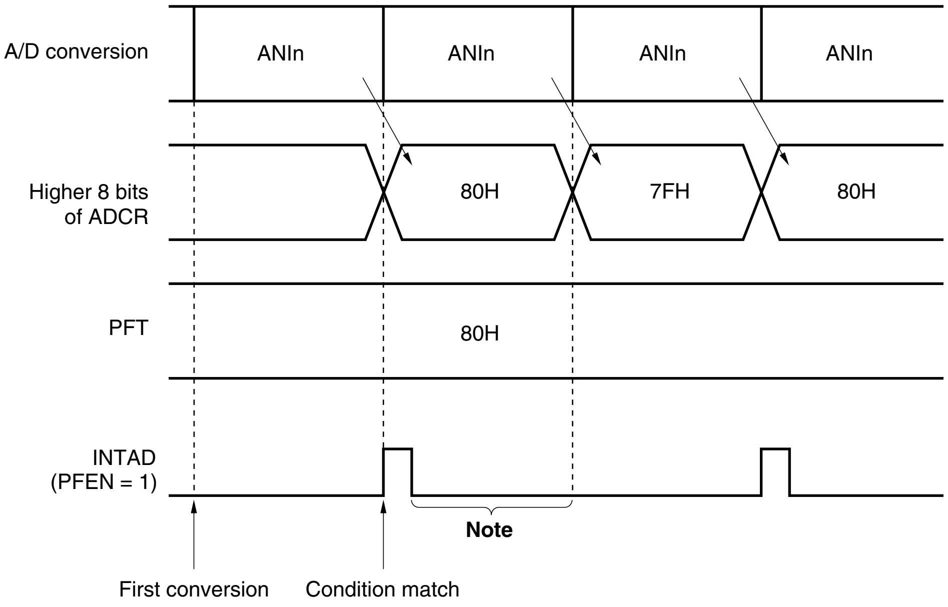
<1> When PFEN = 1 and PFCM = 0

The higher 8 bits of ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when the higher 8 bits of ADCR ≥ PFT.

<2> When PFEN = 1 and PFCM = 1

The higher 8 bits of ADCR and PFT values are compared when A/D conversion ends and INTAD is only generated when the higher 8 bits of ADCR < PFT.

Figure 15-14. Power-Fail Detection (When PFEN = 1 and PFCM = 0)



Note If the conversion result is not read before the end of the next conversion after INTAD is output, the result is replaced by the next conversion result.

Remark n = 0 to 7

(3) Setting

The setting methods are described below.

- When used as normal A/D conversion operation

<1> Set bit 0 (ADCS2) of the A/D converter mode register (ADM) to 1.

<2> Select the channel and conversion time using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS) and bits 5 to 1 (FR2 to FR0, ADHS1, ADHS0) of ADM.

<3> Set bit 7 (ADCS) of ADM to 1 to start the A/D conversion.

<4> An interrupt request signal (INTAD) is generated.

<5> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<Change the channel>

<6> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS to start the A/D conversion.

<7> An interrupt request signal (INTAD) is generated.

<8> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<Complete A/D conversion>

<9> Clear ADCS to 0.

<10> Clear ADCS2 to 0.

Cautions 1. Make sure the period of <1> to <3> is 1 μs or more.

2. It is no problem if the order of <1> and <2> is reversed.

3. <1> can be omitted. However, do not use the first conversion result after <3> in this case.

4. The period from <4> to <7> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, ADHS1, ADHS0) of ADM. The period from <6> to <7> is the conversion time set using FR2 to FR0, ADHS1, and ADHS0.

- When used as power-fail detection function

<1> Set bit 7 (PFEN) of the power-fail comparison mode register (PFM).

<2> Set power-fail comparison condition using bit 6 (PFCM) of PFM.

<3> Set bit 0 (ADCS2) of the A/D converter mode register (ADM) to 1.

<4> Select the channel and conversion time using bits 2 to 0 (ADS2 to ADS0) of the analog input channel specification register (ADS) and bits 5 to 1 (FR2 to FR0, ADHS1, ADHS0) of ADM.

<5> Set a threshold value to the power-fail comparison threshold register (PFT).

<6> Set bit 7 (ADCS) of ADM to 1.

<7> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<8> The higher 8 bits of ADCR and PFT are compared and an interrupt request signal (INTAD) is generated if the conditions match.

<Change the channel>

<9> Change the channel using bits 2 to 0 (ADS2 to ADS0) of ADS.

<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR).

<11> The higher 8 bits of ADCR and the power-fail comparison threshold register (PFT) are compared and an interrupt request signal (INTAD) is generated if the conditions match.

<Complete A/D conversion>

<12> Clear ADCS to 0.

<13> Clear ADCS2 to 0.

Cautions 1. Make sure the period of <3> to <6> is 1 μs or more.

2. It is no problem if the order of <3>, <4>, and <5> is changed.

3. <3> must not be omitted if the power-fail detection function is used.

4. The period from <7> to <11> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, ADHS1, ADHS0) of ADM. The period from <9> to <11> is the conversion time set using FR2 to FR0, ADHS1, and ADHS0.

Remark Regardless of the select mode and scan mode, a compare operation is always performed for all the A/D conversion results when the power fail detection function is enabled.

15.6 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$$1\text{LSB} = 1/2^{10} = 1/1024$$
$$= 0.098\%\text{FSR}$$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 15-15. Overall Error

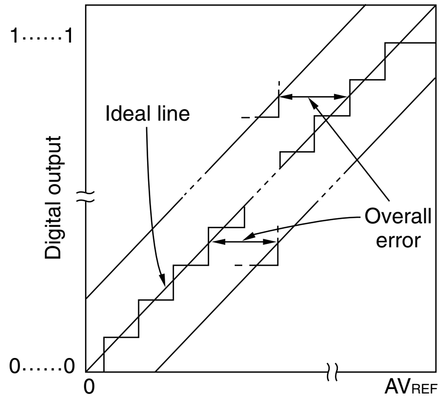
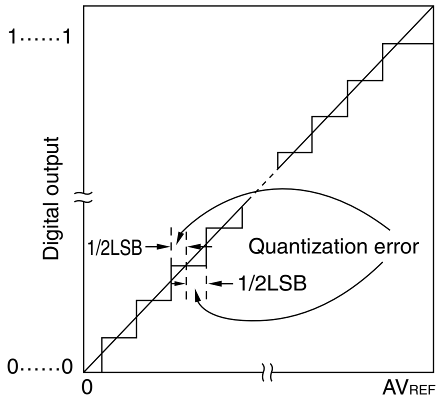


Figure 15-16. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0......000 to 0......001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1......110 to 1......111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 15-17. Zero-Scale Error

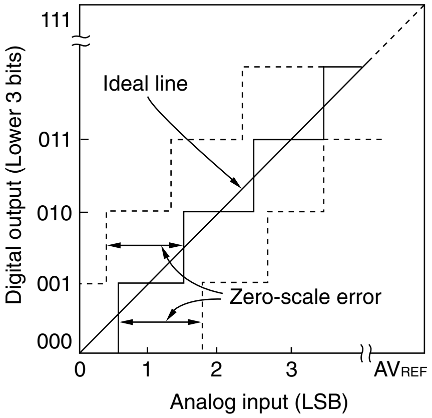


Figure 15-18. Full-Scale Error

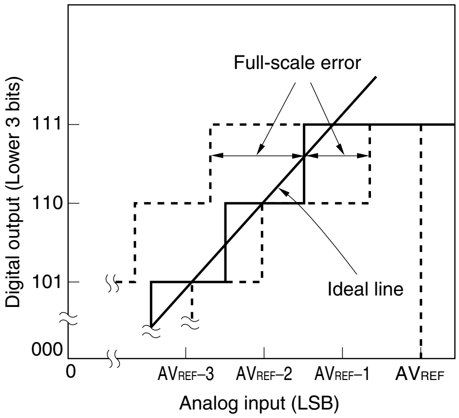


Figure 15-19. Integral Linearity Error

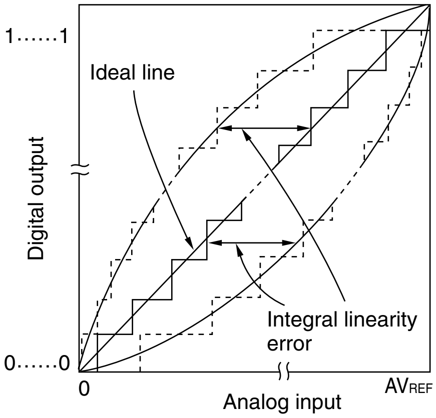
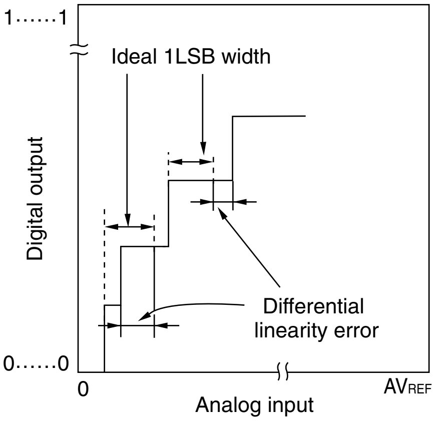


Figure 15-20. Differential Linearity Error



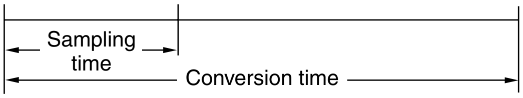
(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



15.7 Cautions for A/D Converter

(1) Operating current in standby mode

The A/D converter stops operating in the standby mode. At this time, the operating current can be reduced by clearing bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 (see **Figure 15-2**).

(2) Input range of ANI0 to ANI7

Observe the rated range of the ANI0 to ANI7 input voltage. If a voltage of AV_{REF} or higher and AV_{SS} or lower (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR) write and ADCR read by instruction upon the end of conversion

ADCR read has priority. After the read operation, the new conversion result is written to ADCR.

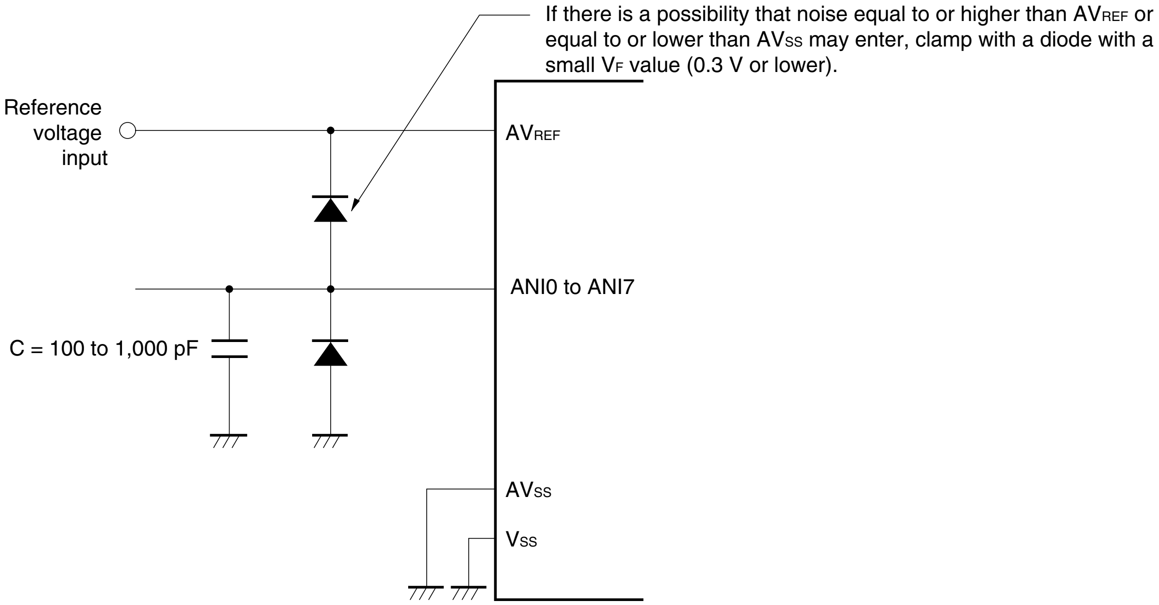
<2> Conflict between ADCR write and A/D converter mode register (ADM) write or analog input channel specification register (ADS) write upon the end of conversion

ADM or ADS write has priority. ADCR write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REF} pin and pins ANI0 to ANI7. Because the effect increases in proportion to the output impedance of the analog input source, it is recommended that a capacitor be connected externally, as shown in Figure 15-21, to reduce noise.

Figure 15-21. Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREF or equal to or lower than AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREF
ANI0 to ANI7
C = 100 to 1,000 pF
AVSS
VSS

(5) ANI0/P20 to ANI7/P27

<1> The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27).
When A/D conversion is performed with any of ANI0 to ANI7 selected, do not access port 2 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a digital pulse is applied to the pins adjacent to the pins currently used for A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, do not apply a pulse to the pins adjacent to the pin undergoing A/D conversion.

(6) Input impedance of ANI0 to ANI7 pins

In this A/D converter, the internal sampling capacitor is charged and sampling is performed.

Since only the leakage current flows other than during sampling and the current for charging the capacitor also flows during sampling, the input impedance fluctuates and has no meaning.

To perform sufficient sampling, however, it is recommended to make the output impedance of the analog input source 10 kΩ or lower, or attach a capacitor of around 100 pF to the ANI0 to ANI7 pins (see **Figure 15-21**).

(7) AV_{REF} pin input impedance

A series resistor string of several tens of 10 kΩ is connected between the AV_{REF} and AV_{SS} pins.

Therefore, if the output impedance of the reference voltage source is high, this will result in a series connection to the series resistor string between the AV_{REF} and AV_{SS} pins, resulting in a large reference voltage error.

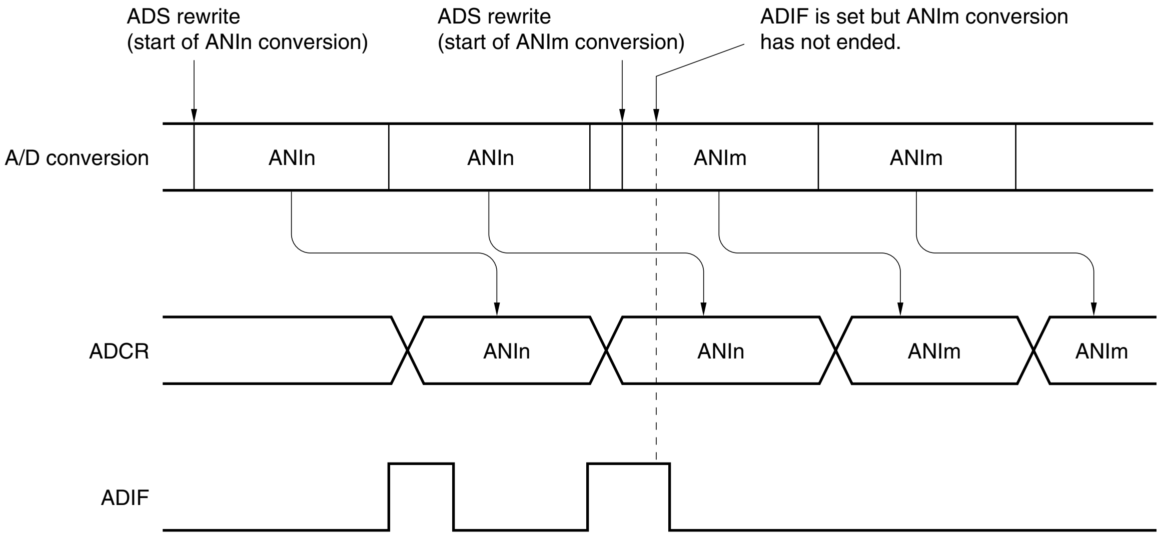
(8) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.

Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF is set despite the fact A/D conversion for the post-change analog input has not ended.

When A/D conversion is stopped and then resumed, clear ADIF before the A/D conversion operation is resumed.

Figure 15-22. Timing of A/D Conversion End Interrupt Request Generation


ADS rewrite
(start of ANIn conversion)
ADS rewrite
(start of ANIm conversion)
ADIF is set but ANIm conversion has not ended.
A/D conversion
ANIn
ANIn
ANIm
ANIm
ADCR
ANIn
ANIn
ANIm
ANIm
ADIF

Remarks 1. n = 0 to 7

2. m = 0 to 7

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCS2 bit was set to 1, or if the ADCS bit is set to 1 with the ADCS2 bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(10) A/D conversion result register (ADCR) read operation

When a write operation is performed to the A/D converter mode register (ADM) and analog input channel specification register (ADS), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM and ADS. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 15-23. Internal Equivalent Circuit of ANIn Pin

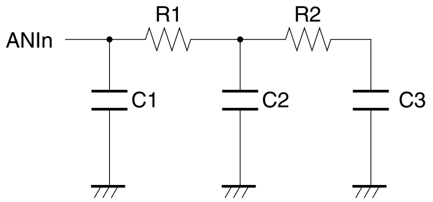


Table 15-4. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REF}	R1	R2	C1	C2	C3
4.5 V	4 kΩ	2.7 kΩ	8 pF	1.4 pF	0.6 pF

Remarks 1. The resistance and capacitance values shown in Table 15-4 are not guaranteed values.

2. n = 0 to 7

CHAPTER 16 SERIAL INTERFACE UART00

16.1 Functions of Serial Interface UART00

Serial interface UART00 has the following two modes.

(1) Operation stop mode

This mode is used when serial communication is not executed and can enable a reduction in the power consumption.

For details, see **16.4.1 Operation stop mode**.

(2) Asynchronous serial interface (UART) mode

The functions of this mode are outlined below.

For details, see **16.4.2 Asynchronous serial interface (UART) mode** and **16.4.3 Dedicated baud rate generator**.

- Two-pin configuration TxD00: Transmit data output pin
  RxD00: Receive data input pin
- Length of communication data can be selected from 7 or 8 bits.
- Dedicated on-chip 5-bit baud rate generator allowing any baud rate to be set
- Transmission and reception can be performed independently.
- Four operating clock inputs selectable
- Fixed to LSB-first communication

Cautions 1. If source clock to serial interface UART00 is not stopped (e.g., in the HALT mode), normal operation continues. If source clock to serial interface UART00 is stopped (e.g., in the STOP mode), each register stops operating, and holds the value immediately before clock supply was stopped. The TxD00 pin also holds the value immediately before clock supply was stopped and outputs it. However, the operation is not guaranteed after clock supply is resumed. Therefore, reset the circuit so that POWER00 = 0, RXE00 = 0, and TXE00 = 0.

2. Set POWER00 = 1 and then set TXE00 = 1 (transmission) or RXE00 = 1 (reception) to start communication.

3. TXE00 and RXE00 are synchronized by the base clock (f_{XCLK0}) set by BRGC00. To enable transmission or reception again, set TXE00 or RXE00 to 1 at least two clocks of base clock after TXE00 or RXE00 has been cleared to 0. If TXE00 or RXE00 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

<R> **4. Set transmit data to TXS00 at least two base clock (f_{XCLK0}) after setting TXE00 = 1.**

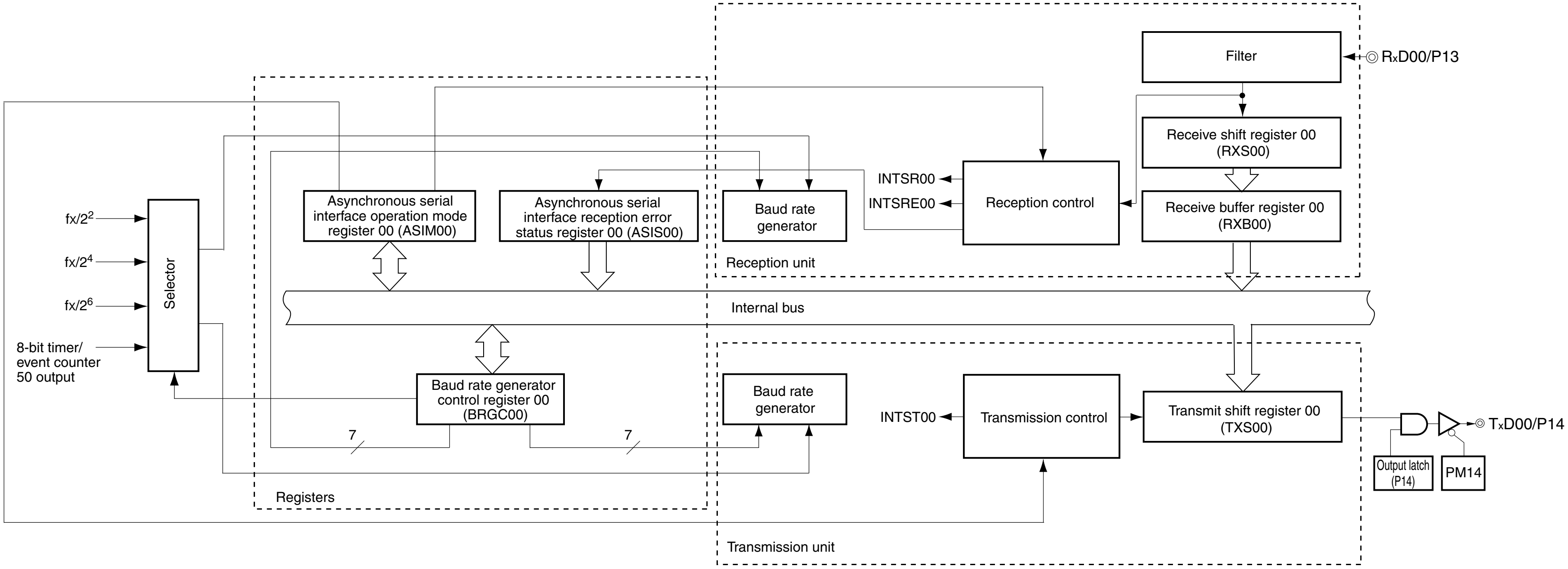
16.2 Configuration of Serial Interface UART00

Serial interface UART00 consists of the following hardware.

Table 16-1. Configuration of Serial Interface UART00

Item	Configuration
Registers	Receive buffer register 00 (RXB00) Receive shift register 00 (RXS00) Transmit shift register 00 (TXS00)
Control registers	Asynchronous serial interface operation mode register 00 (ASIM00) Asynchronous serial interface reception error status register 00 (ASIS00) Baud rate generator control register 00 (BRGC00) Port mode register 1 (PM1) Port register 1 (P1)

Figure 16-1. Block Diagram of Serial Interface UART00


Filter
RxD00/P13
Receive shift register 00 (RXS00)
Receive buffer register 00 (RXB00)
Reception control
INTSR00
INTSRE00
Baud rate generator
Reception unit
Asynchronous serial interface operation mode register 00 (ASIM00)
Asynchronous serial interface reception error status register 00 (ASIS00)
fx/2^2
fx/2^4
fx/2^6
8-bit timer/ event counter 50 output
Selector
Internal bus
Baud rate generator control register 00 (BRGC00)
7
7
Registers
Baud rate generator
Transmission control
INTST00
Transmit shift register 00 (TXS00)
TxD00/P14
Output latch (P14)
PM14
Transmission unit

(1) Receive buffer register 00 (RXB00)

This 8-bit register stores parallel data converted by receive shift register 00 (RXS00).

Each time 1 byte of data has been received, new receive data is transferred to this register from receive shift register 00 (RXS00).

If the data length is set to 7 bits the receive data is transferred to bits 0 to 6 of RXB00 and the MSB of RXB00 is always 0.

If an overrun error (OVE00) occurs, the receive data is not transferred to RXB00.

RXB00 can be read by an 8-bit memory manipulation instruction. No data can be written to this register.

$\overline{\text{RESET}}$ input or POWER00 = 0 sets this register to FFH.

(2) Receive shift register 00 (RXS00)

This register converts the serial data input to the RxD00 pin into parallel data.

RXS00 cannot be directly manipulated by a program.

(3) Transmit shift register 00 (TXS00)

This register is used to set transmit data. Transmission is started when data is written to TXS00, and serial data is transmitted from the TxD00 pin.

TXS00 can be written by an 8-bit memory manipulation instruction. This register cannot be read.

$\overline{\text{RESET}}$ input, POWER00 = 0, or TXE00 = 0 sets this register to FFH.

<R> **Cautions 1. Set transmit data to TXS00 at least two base clock (f_{XCLK0}) after setting TXE00 = 1.**

2. Do not write the next transmit data to TXS00 before the transmission completion interrupt signal (INTST00) is generated.

16.3 Registers Controlling Serial Interface UART00

Serial interface UART00 is controlled by the following five registers.

- Asynchronous serial interface operation mode register 00 (ASIM00)
- Asynchronous serial interface reception error status register 00 (ASIS00)
- Baud rate generator control register 00 (BRGC00)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Asynchronous serial interface operation mode register 00 (ASIM00)

This 8-bit register controls the serial communication operations of serial interface UART00.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to 01H.

Figure 16-2. Format of Asynchronous Serial Interface Operation Mode Register 00 (ASIM00) (1/2)

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM00	POWER00	TXE00	RXE00	PS001	PS000	CL00	SL00	1

POWER00	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit [Note 2].
1	Enables operation of the internal operation clock.

TXE00	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).
1	Enables transmission.

RXE00	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).
1	Enables reception.

Notes 1. The input from the RxD00 pin is fixed to high level when POWER00 = 0.

2. Asynchronous serial interface reception error status register 00 (ASIS00), transmit shift register 00 (TXS00), and receive buffer register 00 (RXB00) are reset.

Figure 16-2. Format of Asynchronous Serial Interface Operation Mode Register 00 (ASIM00) (2/2)

PS001	PS000	Transmission operation	Reception operation
0	0	Does not output parity bit.	Reception without parity
0	1	Outputs 0 parity.	Reception as 0 parity[Note]
1	0	Outputs odd parity.	Judges as odd parity.
1	1	Outputs even parity.	Judges as even parity.

CL00	Specifies character length of transmit/receive data
0	Character length of data = 7 bits
1	Character length of data = 8 bits

SL00	Specifies number of stop bits of transmit data
0	Number of stop bits = 1
1	Number of stop bits = 2

Note If “reception as 0 parity” is selected, the parity is not judged. Therefore, bit 2 (PE00) of asynchronous serial interface reception error status register 00 (ASIS00) is not set and the error interrupt does not occur.

Cautions 1. At startup, set POWER00 to 1 and then set TXE00 to 1. To stop the operation, clear TXE00 to 0, and then clear POWER00 to 0.

2. At startup, set POWER00 to 1 and then set RXE00 to 1. To stop the operation, clear RXE00 to 0, and then clear POWER00 to 0.

3. Set POWER00 to 1 and then set RXE00 to 1 while a high level is input to the RxD00 pin. If POWER00 is set to 1 and RXE00 is set to 1 while a low level is input, reception is started.

4. TXE00 and RXE00 are synchronized by the base clock (f_{XCLK0}) set by BRGC00. To enable transmission or reception again, set TXE00 or RXE00 to 1 at least two clocks of base clock after TXE00 or RXE00 has been cleared to 0. If TXE00 or RXE00 is set within two clocks of base clock, the transmission circuit or reception circuit may not be initialized.

<R> **5. Set transmit data to TXS00 at least two base clock (f_{XCLK0}) after setting TXE00 = 1.**

6. Clear the TXE00 and RXE00 bits to 0 before rewriting the PS001, PS000, and CL00 bits.

7. Make sure that TXE00 = 0 when rewriting the SL00 bit. Reception is always performed with “number of stop bits = 1”, and therefore, is not affected by the set value of the SL00 bit.

8. Be sure to set bit 0 to 1.

(2) Asynchronous serial interface reception error status register 00 (ASIS00)

This register indicates an error status on completion of reception by serial interface UART00. It includes three error flag bits (PE00, FE00, OVE00).

This register is read-only by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, bit 7 (POWER00) of ASIM00 = 0, or bit 5 (RXE00) of ASIM00 = 0 clears this register to 00H. And reading of this register also clears this register to 00H.

Figure 16-3. Format of Asynchronous Serial Interface Reception Error Status Register 00 (ASIS00)

Address: FF73H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ASIS00	0	0	0	0	0	PE00	FE00	OVE00

PE00	Status flag indicating parity error
0	If POWER00 = 0 and RXE00 = 0, or if ASIS00 register is read.
1	If the parity of transmit data does not match the parity bit on completion of reception.

FE00	Status flag indicating framing error
0	If POWER00 = 0 and RXE00 = 0, or if ASIS00 register is read.
1	If the stop bit is not detected on completion of reception.

OVE00	Status flag indicating overrun error
0	If POWER00 = 0 and RXE00 = 0, or if ASIS00 register is read.
1	If receive data is set to the RXB00 register and the next reception operation is completed before the data is read.

Cautions 1. The operation of the PE00 bit differs depending on the set values of the PS001 and PS000 bits of asynchronous serial interface operation mode register 00 (ASIM00).

2. Only the first bit of the receive data is checked as the stop bit, regardless of the number of stop bits.

3. If an overrun error occurs, the next receive data is not written to receive buffer register 00 (RXB00) but discarded.

4. If data is read from ASIS00, a wait cycle is generated. For details, see CHAPTER 30 CAUTIONS FOR WAIT.

(3) Baud rate generator control register 00 (BRGC00)

This register selects the base clock of serial interface UART00 and the division value of the 5-bit counter.

BRGC00 can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 1FH.

Figure 16-4. Format of Baud Rate Generator Control Register 00 (BRGC00)

Address: FF71H After reset: 1FH R/W

Symbol	7	6	5	4	3	2	1	0
BRGC00	TPS001	TPS000	0	MDL004	MDL003	MDL002	MDL001	MDL000

TPS001	TPS000	Base clock (f_{XCLK0}) selection [Note 1]
0	0	TM50 output [Note 2]
0	1	$f_X/2^2$ (5 MHz)
1	0	$f_X/2^4$ (1.25 MHz)
1	1	$f_X/2^6$ (312.5 kHz)

MDL004	MDL003	MDL002	MDL001	MDL000	k	Selection of 5-bit counter output clock
0	0	×	×	×	×	Setting prohibited
0	1	0	0	0	8	$f_{XCLK0}/8$
0	1	0	0	1	9	$f_{XCLK0}/9$
0	1	0	1	0	10	$f_{XCLK0}/10$
• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •	• • • • •
1	1	0	1	1	27	$f_{XCLK0}/27$
1	1	1	0	0	28	$f_{XCLK0}/28$
1	1	1	0	1	29	$f_{XCLK0}/29$
1	1	1	1	0	30	$f_{XCLK0}/30$
1	1	1	1	1	31	$f_{XCLK0}/31$

Notes 1. Be sure to set the base clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Base clock ≤ 10 MHz

2. When the TM50 output is selected as the count clock, observe the following.

- PWM mode (TMC506 = 1)

  Set the clock so that the duty will be 50% and start the operation of 8-bit timer/event counter 50 in advance.

- Clear & start mode entered on match of TM50 and CR50 (TMC506 = 0)

  Enable the timer F/F inversion operation (TMC501 = 1) and start the operation of 8-bit timer/event counter 50 in advance.

It is not necessary to enable the TO50 pin as a timer output pin (bit 00 (TOE50) of the TMC register may be 0 or 1), regardless which mode.

Cautions 1. When the internal oscillation clock is selected as the source clock to the CPU, the clock of the internal oscillator is divided and supplied as the count clock. If the base clock is the internal oscillation clock, the operation of serial interface UART00 is not guaranteed.

2. Make sure that bit 6 (TXE00) and bit 5 (RXE00) of the ASIM00 register = 0 when rewriting the MDL004 to MDL000 bits.

3. The baud rate value is the output clock of the 5-bit counter divided by 2.

Remarks 1. f_{XCLK0}: Frequency of base clock selected by the TPS001 and TPS000 bits

2. f_X: X1 input clock oscillation frequency

3. k: Value set by the MDL004 to MDL000 bits (k = 8, 9, 10, ..., 31)

4. ×: Don’t care

5. Figures in parentheses apply to operation at f_X = 20 MHz

6. TMC506: Bit 6 of 8-bit timer mode control register 50 (TMC50)
TMC501: Bit 1 of TMC50

(4) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using the P14/TxD00 pin for serial interface data output, clear PM14 to 0 and set the output latch of P14 to 1.

When using the P13/RxD00 pin for serial interface data input, set PM13 to 1. The output latch of P13 at this time may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to FFH.

Figure 16-5. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

16.4 Operation of Serial Interface UART00

Serial interface UART00 has the following two modes.

- Operation stop mode
- Asynchronous serial interface (UART) mode

16.4.1 Operation stop mode

In this mode, serial communication cannot be executed, thus reducing the power consumption. In addition, the pins can be used as ordinary port pins in this mode. To set the operation stop mode, clear bits 7, 6, and 5 (POWER00, TXE00, and RXE00) of ASIM00 to 0.

(1) Register used

The operation stop mode is set by asynchronous serial interface operation mode register 00 (ASIM00).
ASIM00 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets this register to 01H.

Address: FF70H After reset: 01H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
ASIM00	POWER00	TXE00	RXE00	PS001	PS000	CL00	SL00	1

POWER00	Enables/disables operation of internal operation clock
0[Note 1]	Disables operation of the internal operation clock (fixes the clock to low level) and asynchronously resets the internal circuit[Note 2].

TXE00	Enables/disables transmission
0	Disables transmission (synchronously resets the transmission circuit).

RXE00	Enables/disables reception
0	Disables reception (synchronously resets the reception circuit).

Notes 1. The input from the RxD00 pin is fixed to high level when POWER00 = 0.

2. Asynchronous serial interface reception error status register 00 (ASIS00), transmit shift register 00 (TXS00), and receive buffer register 00 (RXB00) are reset.

Caution Clear POWER00 to 0 after clearing TXE00 and RXE00 to 0 to set the operation stop mode. To start the operation, set POWER00 to 1, and then set TXE00 and RXE00 to 1.

Remark To use the RxD00/P13 and TxD00/P14 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS**.

16.4.2 Asynchronous serial interface (UART) mode

In this mode, 1-byte data is transmitted/received following a start bit, and a full-duplex operation can be performed.

A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

(1) Registers used

- Asynchronous serial interface operation mode register 00 (ASIM00)
- Asynchronous serial interface reception error status register 00 (ASIS00)
- Baud rate generator control register 00 (BRGC00)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the UART mode is as follows.

<1> Set the BRGC00 register (see **Figure 16-4**).
<2> Set bits 1 to 4 (SL00, CL00, PS000, and PS001) of the ASIM00 register (see **Figure 16-2**).
<3> Set bit 7 (POWER00) of the ASIM00 register to 1.
<4> Set bit 6 (TXE00) of the ASIM00 register to 1. → Transmission is enabled.
Set bit 5 (RXE00) of the ASIM00 register to 1. → Reception is enabled.
<R> <5> Write data to the TXS00 register at least two clock after setting <4>. → Data transmission is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 16-2. Relationship Between Register Settings and Pins

POWER00	TXE00	RXE00	PM14	P14	PM13	P13	UART00 Operation	Pin Function	
								TxD00/ P14	RxD00/P13
0	0	0	×[Note]	×[Note]	×[Note]	×[Note]	Stop	P14	P13
1	0	1	×[Note]	×[Note]	1	×	Reception	P14	RxD00
	1	0	0	1	×[Note]	×[Note]	Transmission	TxD00	P13
	1	1	0	1	1	×	Transmission/ reception	TxD00	RxD00

Note Can be set as port function.

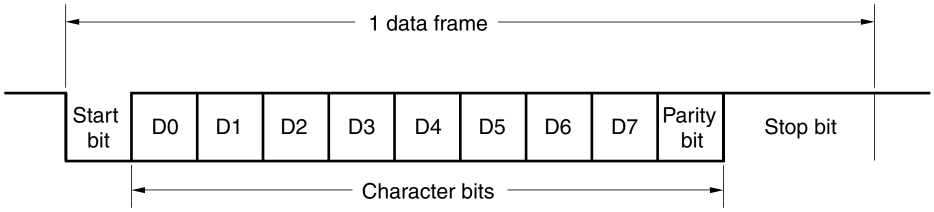
Remark	
×:	don't care
POWER00:	Bit 7 of asynchronous serial interface operation mode register 00 (ASIM00)
TXE00:	Bit 6 of ASIM00
RXE00:	Bit 5 of ASIM00
PM1×:	Port mode register
P1×:	Port output latch

(2) **Communication operation**

(a) **Format and waveform example of normal transmit/receive data**

Figures 16-6 and 16-7 show the format and waveform example of the normal transmit/receive data.

Figure 16-6. Format of Normal UART Transmit/Receive Data



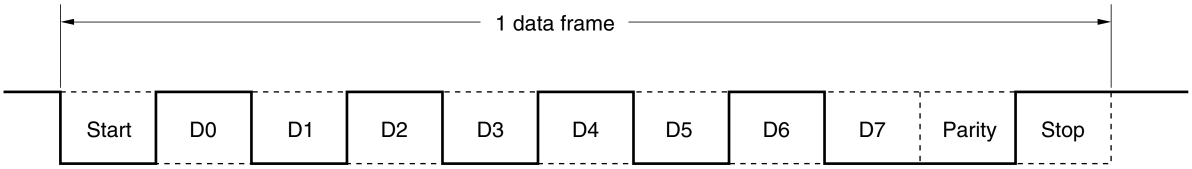
One data frame consists of the following bits.

- Start bit ... 1 bit
- Character bits ... 7 or 8 bits (LSB first)
- Parity bit ... Even parity, odd parity, 0 parity, or no parity
- Stop bit ... 1 or 2 bits

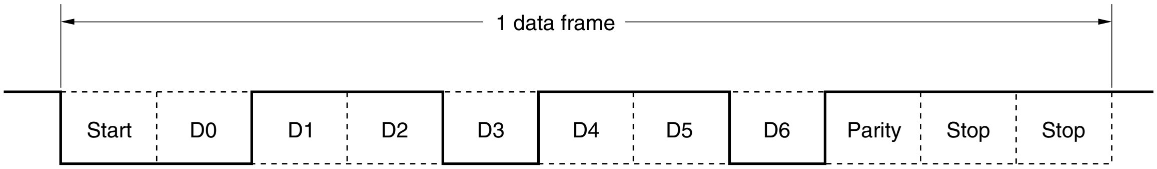
The character bit length, parity, and stop bit length in one data frame are specified by asynchronous serial interface operation mode register 00 (ASIM00).

Figure 16-7. Example of Normal UART Transmit/Receive Data Waveform

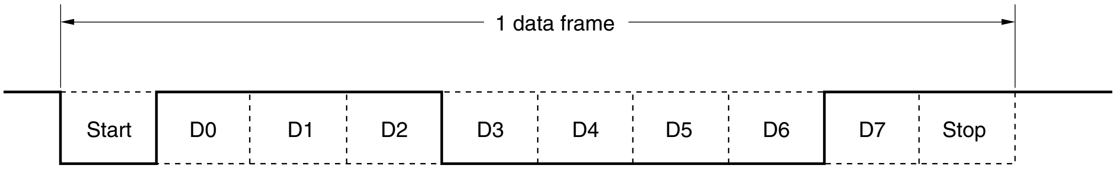
1. **Data length: 8 bits, Parity: Even parity, Stop bit: 1 bit, Communication data: 55H**



2. **Data length: 7 bits, Parity: Odd parity, Stop bit: 2 bits, Communication data: 36H**



3. **Data length: 8 bits, Parity: None, Stop bit: 1 bit, Communication data: 87H**



(b) Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmission and reception sides. With even parity and odd parity, a 1-bit (odd number) error can be detected. With zero parity and no parity, an error cannot be detected.

(i) Even parity

- Transmission

  Transmit data, including the parity bit, is controlled so that the number of bits that are “1” is even.
  The value of the parity bit is as follows.

  If transmit data has an odd number of bits that are “1”: 1
  If transmit data has an even number of bits that are “1”: 0

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is odd, a parity error occurs.

(ii) Odd parity

- Transmission

  Unlike even parity, transmit data, including the parity bit, is controlled so that the number of bits that are “1” is odd.

  If transmit data has an odd number of bits that are “1”: 0
  If transmit data has an even number of bits that are “1”: 1

- Reception

  The number of bits that are “1” in the receive data, including the parity bit, is counted. If it is even, a parity error occurs.

(iii) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data.
The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is “0” or “1”.

(iv) No parity

No parity bit is appended to the transmit data.
Reception is performed assuming that there is no parity bit when data is received. Because there is no parity bit, a parity error does not occur.

(c) Transmission

The TxD00 pin outputs a high level when bit 7 (POWER00) of asynchronous serial interface operation mode register 00 (ASIM00) is set to 1. If bit 6 (TXE00) of ASIM00 is then set to 1, transmission is enabled. Transmission can be started by writing transmit data to transmit shift register 00 (TXS00) at least two base clock (f_{XCLK0}) after setting TXE00 = 1. The start bit, parity bit, and stop bit are automatically appended to the data.

When transmission is started, the start bit is output from the TxD00 pin, followed by the rest of the data in order starting from the LSB. When transmission is completed, the parity and stop bits set by ASIM00 are appended and a transmission completion interrupt request (INTST00) is generated.

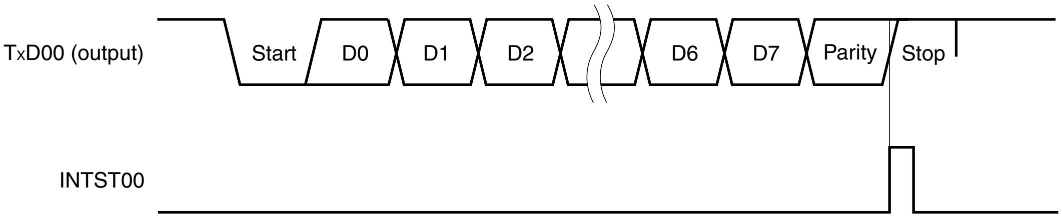
Transmission is stopped until the data to be transmitted next is written to TXS00.

Figure 16-8 shows the timing of the transmission completion interrupt request (INTST00). This interrupt occurs as soon as the last stop bit has been output.

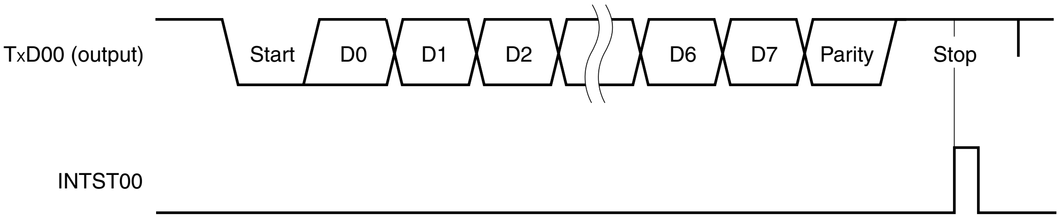
Caution After transmit data is written to TXS00, do not write the next transmit data before the transmission completion interrupt signal (INTST00) is generated.

Figure 16-8. Transmission Completion Interrupt Request Timing

1. Stop bit length: 1



2. Stop bit length: 2



(d) Reception

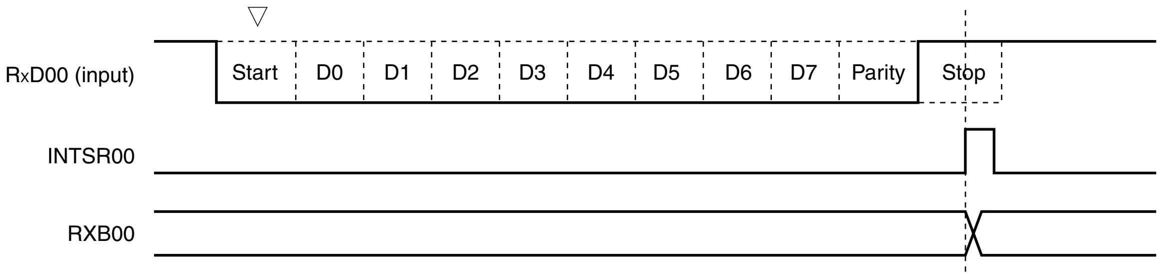
Reception is enabled and the RxD00 pin input is sampled when bit 7 (POWER00) of asynchronous serial interface operation mode register 00 (ASIM00) is set to 1 and then bit 5 (RXE00) of ASIM00 is set to 1.

The 5-bit counter of the baud rate generator starts counting when the falling edge of the RxD00 pin input is detected. When the set value of baud rate generator control register 00 (BRGC00) has been counted, the RxD00 pin input is sampled again (▽ in Figure 16-9). If the RxD00 pin is low level at this time, it is recognized as a start bit.

When the start bit is detected, reception is started, and serial data is sequentially stored in receive shift register 00 (RXS00) at the set baud rate. When the stop bit has been received, the reception completion interrupt (INTSR00) is generated and the data of RXS00 is written to receive buffer register 00 (RXB00). If an overrun error (OVE00) occurs, however, the receive data is not written to RXB00.

Even if a parity error (PE00) occurs while reception is in progress, reception continues to the reception position of the stop bit, and an error interrupt (INTSRE00) is generated after completion of reception.

Figure 16-9. Reception Completion Interrupt Request Timing



Cautions 1. Be sure to read receive buffer register 00 (RXB00) even if a reception error occurs. Otherwise, an overrun error will occur when the next data is received, and the reception error status will persist.

2. Reception is always performed with the "number of stop bits = 1". The second stop bit is ignored.

3. Be sure to read asynchronous serial interface reception error status register 00 (ASIS00) before reading RXB00.

(e) Reception error

Three types of errors may occur during reception: a parity error, framing error, or overrun error. If the error flag of asynchronous serial interface reception error status register 00 (ASIS00) is set as a result of data reception, a reception error interrupt request (INTSRE00) is generated.

Which error has occurred during reception can be identified by reading the contents of ASIS00 in the reception error interrupt servicing (INTSRE00) (see **Figure 16-3**).

The contents of ASIS00 are reset to 0 when ASIS00 is read.

Table 16-3. Cause of Reception Error

Reception Error	Cause
Parity error	The parity specified for transmission does not match the parity of the receive data.
Framing error	Stop bit is not detected.
Overrun error	Reception of the next data is completed before data is read from receive buffer register 00 (RXB00).

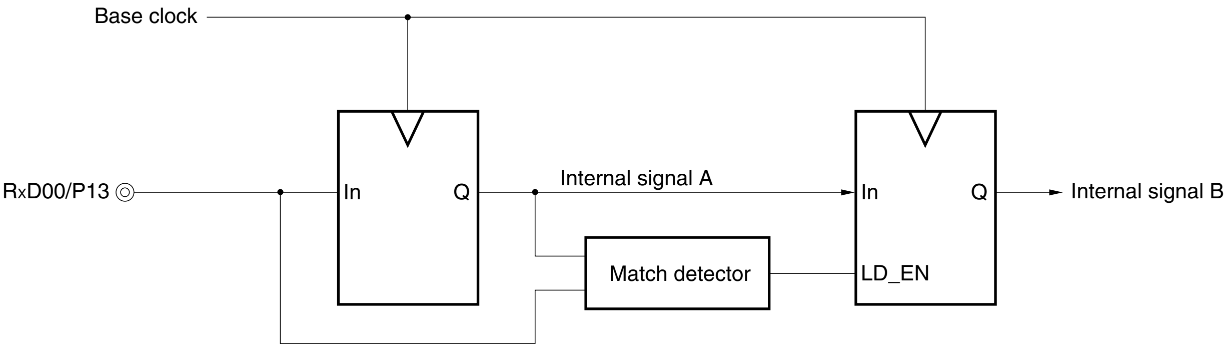
(f) Noise filter of receive data

The RxD00 signal is sampled using the base clock output by the prescaler block.

If two sampled values are the same, the output of the match detector changes, and the data is sampled as input data.

Because the circuit is configured as shown in Figure 16-10, the internal processing of the reception operation is delayed by two clocks from the external signal status.

Figure 16-10. Noise Filter Circuit



16.4.3 Dedicated baud rate generator

The dedicated baud rate generator consists of a source clock selector and a 5-bit programmable counter, and generates a serial clock for transmission/reception of UART00.

Separate 5-bit counters are provided for transmission and reception.

(1) Configuration of baud rate generator

- Base clock

  The clock selected by bits 7 and 6 (TPS001 and TPS000) of baud rate generator control register 00 (BRGC00) is supplied to each module when bit 7 (POWER00) of asynchronous serial interface operation mode register 00 (ASIM00) is 1. This clock is called the base clock and its frequency is called f_{XCLK0}. The base clock is fixed to low level when POWER00 = 0.

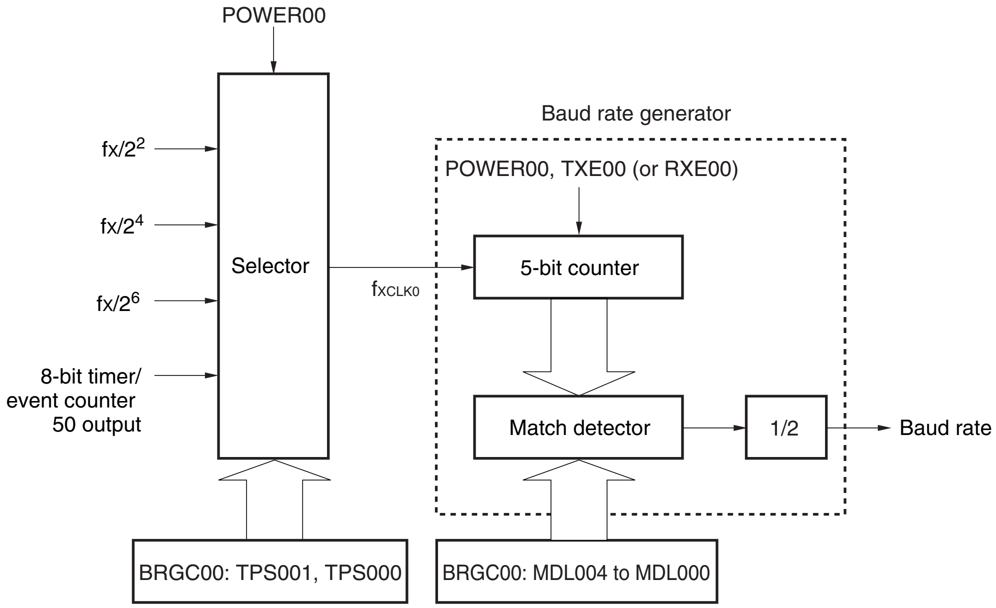
- Transmission counter

  This counter stops operation, cleared to 0, when bit 7 (POWER00) or bit 6 (TXE00) of asynchronous serial interface operation mode register 00 (ASIM00) is 0.
  It starts counting when POWER00 = 1 and TXE00 = 1.
  The counter is cleared to 0 when the first data transmitted is written to transmit shift register 00 (TXS00).

- Reception counter

  This counter stops operation, cleared to 0, when bit 7 (POWER00) or bit 5 (RXE00) of asynchronous serial interface operation mode register 00 (ASIM00) is 0.
  It starts counting when the start bit has been detected.
  The counter stops operation after one frame has been received, until the next start bit is detected.

Figure 16-11. Configuration of Baud Rate Generator



Remark		
	POWER00:	Bit 7 of asynchronous serial interface operation mode register 00 (ASIM00)
	TXE00:	Bit 6 of ASIM00
	RXE00:	Bit 5 of ASIM00
	BRGC00:	Baud rate generator control register 00

(2) Generation of serial clock

A serial clock can be generated by using baud rate generator control register 00 (BRGC00).

Select the clock to be input to the 5-bit counter by using bits 7 and 6 (TPS001 and TPS000) of BRGC00.

Bits 4 to 0 (MDL004 to MDL000) of BRGC00 can be used to select the division value of the 5-bit counter.

(a) Baud rate

The baud rate can be calculated by the following expression.

- Baud rate = $\frac{f_{XCLK0}}{2 \times k}$ [bps]

f_{XCLK0}: Frequency of base clock selected by the TPS001 and TPS000 bits of the BRGC00 register

k: Value set by the MDL004 to MDL000 bits of the BRGC00 register (k = 8, 9, 10, ..., 31)

(b) Error of baud rate

The baud rate error can be calculated by the following expression.

- Error (%) = $\left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100$ [%]

Cautions 1. Keep the baud rate error during transmission to within the permissible error range at the reception destination.

2. Make sure that the baud rate error during reception satisfies the range shown in (4) Permissible baud rate range during reception.

Example: Frequency of base clock = 2.5 MHz = 2,500,000 Hz

Set value of MDL004 to MDL000 bits of BRGC00 register = 10000B (k = 16)

Target baud rate = 76,800 bps

Baud rate = 2.5 M/(2 × 16)

= 2,500,000/(2 × 16) = 78,125 [bps]

Error = (78,125/76,800 − 1) × 100

= 1.725 [%]

(3) Example of setting baud rate

Table 16-4. Set Data of Baud Rate Generator

Baud Rate [bps]	fX = 20.0 MHz				fX = 16.0 MHz			
	TPS001, TPS000	k	Calculated Value	ERR[%]	TPS001, TPS000	k	Calculated Value	ERR[%]
2400	–	–	–	–	–	–	–	–
4800	–	–	–	–	3	26	4808	0.16
9600	3	16	9766	1.73	3	13	9615	0.16
10400	3	15	10417	0.16	3	12	10417	0.16
19200	3	8	19531	1.73	2	26	19231	0.16
31250	2	20	31250	0	2	16	31250	0
38400	2	16	39063	1.73	2	13	38462	0.16
76800	2	8	78125	1.73	1	26	76923	0.16
115200	1	22	113636	−1.36	1	17	117647	2.12
153600	1	16	156250	1.73	1	13	153846	0.16
230400	1	11	227273	−1.36	–	–	–	–

Remark TPS001, TPS000: Bits 7 and 6 of baud rate generator control register 00 (BRGC00) (setting of base clock (f_{XCLK0}))

k: Value set by the MDL004 to MDL000 bits of BRGC00 (k = 8, 9, 10, ..., 31)

fX: X1 input clock oscillation frequency

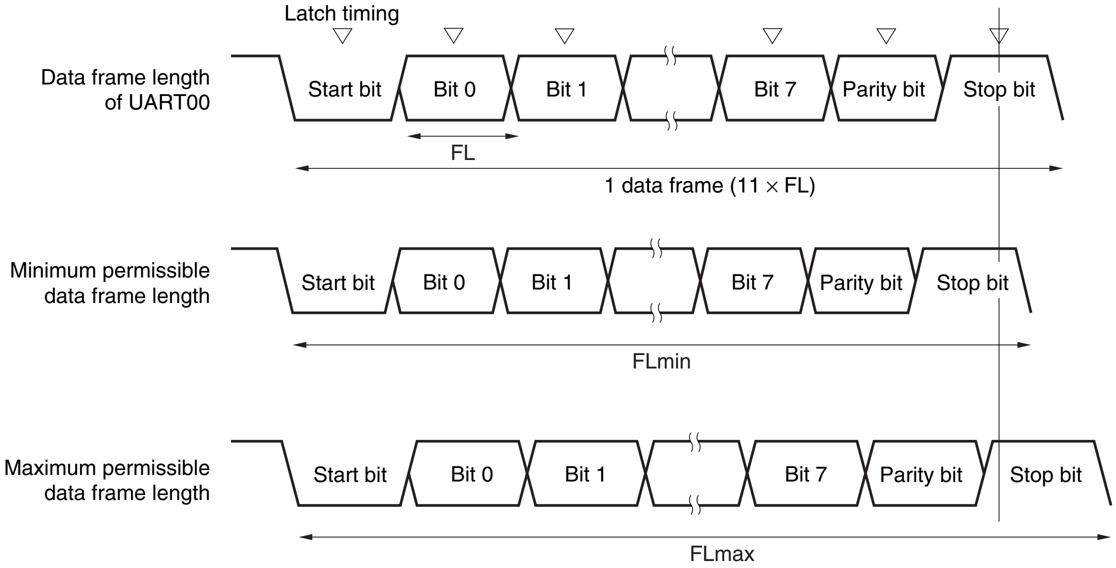
ERR: Baud rate error

(4) Permissible baud rate range during reception

The permissible error from the baud rate at the transmission destination during reception is shown below.

Caution Make sure that the baud rate error during reception is within the permissible error range, by using the calculation expression shown below.

Figure 16-12. Permissible Baud Rate Range During Reception



As shown in Figure 16-12, the latch timing of the receive data is determined by the counter set by baud rate generator control register 00 (BRGC00) after the start bit has been detected. If the last data (stop bit) meets this latch timing, the data can be correctly received.
Assuming that 11-bit data is received, the theoretical values can be calculated as follows.

$$\text{FL} = (\text{Brate})^{-1}$$

Brate: Baud rate of UART00
k: Set value of BRGC00
FL: 1-bit data length
Margin of latch timing: 2 clocks

Minimum permissible data frame length: $\text{FLmin} = 11 \times \text{FL} - \frac{k-2}{2k} \times \text{FL} = \frac{21k+2}{2k}\ \text{FL}$

Therefore, the maximum receivable baud rate at the transmission destination is as follows.

$$\text{BRmax} = (\text{FLmin}/11)^{-1} = \frac{22k}{21k+2}\ \text{Brate}$$

Similarly, the maximum permissible data frame length can be calculated as follows.

$$\frac{10}{11} \times \text{FLmax} = 11 \times \text{FL} - \frac{k+2}{2 \times k} \times \text{FL} = \frac{21k-2}{2 \times k}\ \text{FL}$$

$$\text{FLmax} = \frac{21k-2}{20k}\ \text{FL} \times 11$$

Therefore, the minimum receivable baud rate at the transmission destination is as follows.

$$\text{BRmin} = (\text{FLmax}/11)^{-1} = \frac{20k}{21k-2}\ \text{Brate}$$

The permissible baud rate error between UART00 and the transmission destination can be calculated from the above minimum and maximum baud rate expressions, as follows.

Table 16-5. Maximum/Minimum Permissible Baud Rate Error

Division Ratio (k)	Maximum Permissible Baud Rate Error	Minimum Permissible Baud Rate Error
8	+3.53%	–3.61%
16	+4.14%	–4.19%
24	+4.34%	–4.38%
31	+4.44%	–4.47%

Remarks 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the higher the division ratio (k), the higher the permissible error.

2. k: Set value of BRGC00

CHAPTER 17 SERIAL INTERFACE CSI10

17.1 Functions of Serial Interface CSI10

Serial interface CSI10 has the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

(1) Operation stop mode

This mode is used when serial communication is not performed and can enable a reduction in the power consumption.

For details, see **17.4.1 Operation stop mode**.

(2) 3-wire serial I/O mode (MSB/LSB-first selectable)

This mode is used to communicate 8-bit data using three lines: a serial clock line ($\overline{\text{SCK10}}$) and two serial data lines (SI10 and SO10).

The processing time of data communication can be shortened in the 3-wire serial I/O mode because transmission and reception can be simultaneously executed.

In addition, whether 8-bit data is communicated with the MSB or LSB first can be specified, so this interface can be connected to any device.

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

For details, see **17.4.2 3-wire serial I/O mode**.

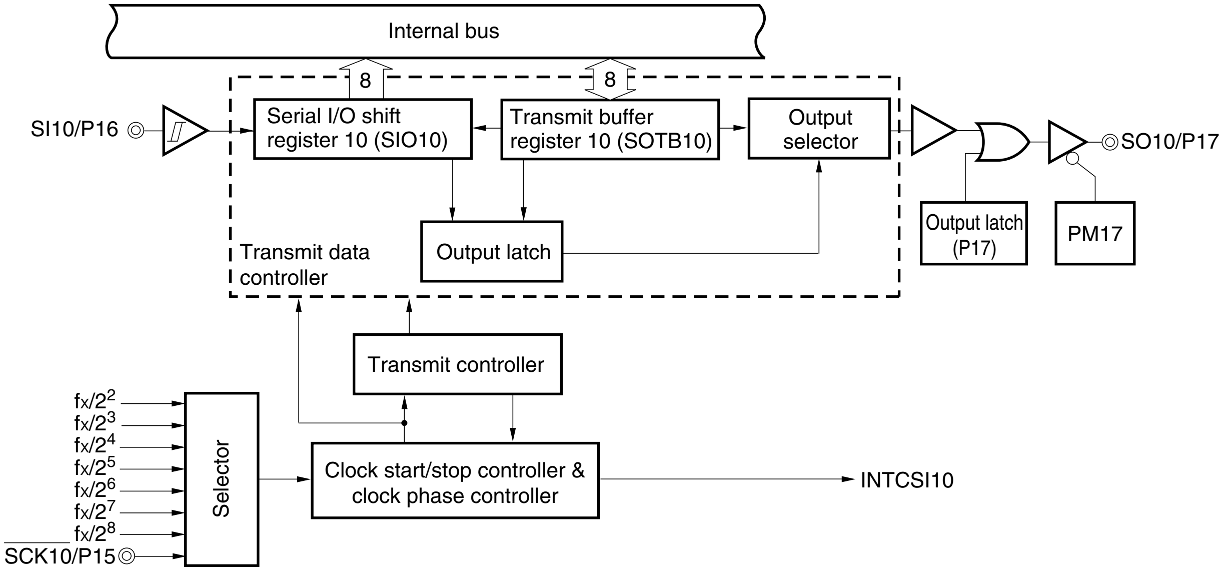
17.2 Configuration of Serial Interface CSI10

Serial interface CSI10 consists of the following hardware.

Table 17-1. Configuration of Serial Interface CSI10

Item	Configuration
Registers	Transmit buffer register 10 (SOTB10) Serial I/O shift register 10 (SIO10) Transmit controller Clock start/stop controller & clock phase controller
Control registers	Serial operation mode register 10 (CSIM10) Serial clock selection register 10 (CSIC10) Port mode register 1 (PM1) Port register 1 (P1)

Figure 17-1. Block Diagram of Serial Interface CSI10



(1) Transmit buffer register 10 (SOTB10)

This register sets the transmit data.

Transmission/reception is started by writing data to SOTB10 when bit 7 (CSIE10) and bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1.

The data written to SOTB10 is converted from parallel data into serial data by serial I/O shift register 10, and output to the serial output pin (SO10).

SOTB10 can be written or read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input makes this register undifined.

Caution Do not access SOTB10 when CSOT10 = 1 (during serial communication).

(2) Serial I/O shift register 10 (SIO10)

This is an 8-bit register that converts data from parallel data into serial data and vice versa.

This register can be read by an 8-bit memory manipulation instruction.

Reception is started by reading data from SIO10 if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

During reception, the data is read from the serial input pin (SI10) to SIO10.

$\overline{\text{RESET}}$ input clears this register to 00H.

Caution Do not access SIO10 when CSOT10 = 1 (during serial communication).

17.3 Registers Controlling Serial Interface CSI10

Serial interface CSI10 is controlled by the following four registers.

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port mode register 1 (PM1)
- Port register 1 (P1)

(1) Serial operation mode register 10 (CSIM10)

CSIM10 is used to select the operation mode and enable or disable operation.

CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 17-2. Format of Serial Operation Mode Register 10 (CSIM10)

Address: FF80H After reset: 00H R/W[Note 1]

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 2] and asynchronously resets the internal circuit[Note 3].
1	Enables operation

TRMD10[Note 4]	Transmit/receive mode control
0[Note 5]	Receive mode (transmission disabled).
1	Transmit/receive mode

DIR10[Note 6]	First bit specification
0	MSB
1	LSB

CSOT10	Communication status flag
0	Communication is stopped.
1	Communication is in progress.

Notes 1. Bit 0 is a read-only bit.

2. When using P15/$\overline{\text{SCK10}}$, P16/SI10, and P17/SO10/FLMD1 as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS** and **Caution 3** of **Figure 17-3**.

3. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

4. Do not rewrite TRMD10 when CSOT10 = 1 (during serial communication).

5. The SO10 output is fixed to the low level when TRMD10 is 0. Reception is started when data is read from SIO10.

6. Do not rewrite DIR10 when CSOT10 = 1 (during serial communication).

Caution Be sure to clear bit 5 to 0.

(2) Serial clock selection register 10 (CSIC10)

This register specifies the timing of the data transmission/reception and sets the serial clock.

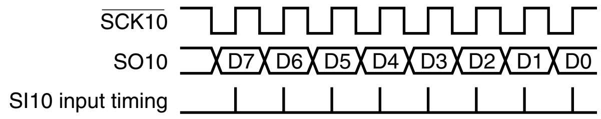
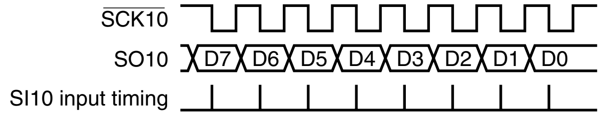
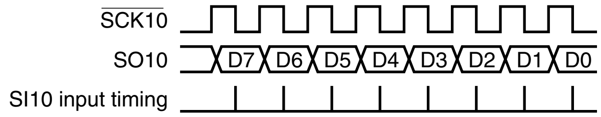
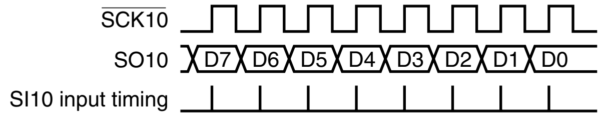
CSIC10 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 17-3. Format of Serial Clock Selection Register 10 (CSIC10)

Address: FF81H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CSIC10	0	0	0	CKP10	DAP10	CKS102	CKS101	CKS100

<table>
<tr><th>CKP10</th><th>DAP10</th><th>Specification of data transmission/reception timing</th><th>Type</th></tr>
<tr><td>0</td><td>0</td><td> </td><td>1</td></tr>
<tr><td>0</td><td>1</td><td> </td><td>2</td></tr>
<tr><td>1</td><td>0</td><td> </td><td>3</td></tr>
<tr><td>1</td><td>1</td><td> </td><td>4</td></tr>
</table>

CKS102	CKS101	CKS100	CSI10 serial clock selection[Note]	Mode
0	0	0	$f_X/2^2$ (5 MHz)	Master mode
0	0	1	$f_X/2^3$ (2.5 MHz)	Master mode
0	1	0	$f_X/2^4$ (1.25 MHz)	Master mode
0	1	1	$f_X/2^5$ (625 kHz)	Master mode
1	0	0	$f_X/2^6$ (312.5 kHz)	Master mode
1	0	1	$f_X/2^7$ (156.25 kHz)	Master mode
1	1	0	$f_X/2^8$ (78.13 kHz)	Master mode
1	1	1	External clock input to $\overline{\text{SCK10}}$	Slave mode

Note Be sure to set the serial clock so that the following condition is satisfied.

- V_{DD} = 4.0 to 5.5 V: Serial clock ≤ 5 MHz

Cautions 1. When the internal oscillation clock is selected as the clock supplied to the CPU, the clock of the internal oscillator is divided and supplied as the serial clock. At this time, the operation of serial interface CSI10 is not guaranteed.

2. Do not write to CSIC10 while CSIE10 = 1 (operation enabled).

3. Clear CKP10 to 0 to use P15/$\overline{\text{SCK10}}$, P16/SI10, and P17/SO10/FLMD1 as general-purpose port pins.

4. The phase type of the data clock is type 1 after reset.

Remarks 1. Figures in parentheses are for operation with fx = 20 MHz

2. fx: X1 input clock oscillation frequency

(3) Port mode register 1 (PM1)

This register sets port 1 input/output in 1-bit units.

When using P15/$\overline{\text{SCK10}}$ as the clock output pin of the serial interface, clear PM15 to 0 and set the output latch of P15 to 1.

When P17/SO10/FLMD1 as the data output pin, clear PM17 and the output latch of P17 to 0.

When using P15/$\overline{\text{SCK10}}$ as the clock input pin of the serial interface, and P16/SI10 as the data input pin, set PM15 and PM16 to 1. At this time, the output latches of P15 and P16 may be 0 or 1.

PM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to FFH.

Figure 17-4. Format of Port Mode Register 1 (PM1)

Address: FF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	P1n pin I/O mode selection (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

17.4 Operation of Serial Interface CSI10

Serial interface CSI10 can be used in the following two modes.

- Operation stop mode
- 3-wire serial I/O mode

17.4.1 Operation stop mode

Serial communication is not executed in this mode. Therefore, the power consumption can be reduced. In addition, the P15/$\overline{\text{SCK10}}$, P16/SI10, and P17/SO10/FLMD1 pins can be used as ordinary I/O port pins in this mode.

(1) Register used

The operation stop mode is set by serial operation mode register 10 (CSIM10).
To set the operation stop mode, clear bit 7 (CSIE10) of CSIM10 to 0.

(a) Serial operation mode register 10 (CSIM10)

CSIM10 can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears CSIM10 to 00H.

Address: FF80H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CSIM10	CSIE10	TRMD10	0	DIR10	0	0	0	CSOT10

CSIE10	Operation control in 3-wire serial I/O mode
0	Disables operation[Note 1] and asynchronously resets the internal circuit[Note 2].

Notes 1. When using the P15/$\overline{\text{SCK10}}$, P16/SI10, and P17/SO10/FLMD1 pins as general-purpose port pins, see **CHAPTER 4 PORT FUNCTIONS** and **Caution 3** of **Figure 17-3**.

2. Bit 0 (CSOT10) of CSIM10 and serial I/O shift register 10 (SIO10) are reset.

17.4.2 3-wire serial I/O mode

The 3-wire serial I/O mode is used for connecting peripheral ICs and display controllers with a clocked serial interface.

In this mode, communication is executed by using three lines: the serial clock ($\overline{\text{SCK10}}$), serial output (SO10), and serial input (SI10) lines.

(1) Registers used

- Serial operation mode register 10 (CSIM10)
- Serial clock selection register 10 (CSIC10)
- Port mode register 1 (PM1)
- Port register 1 (P1)

The basic procedure of setting an operation in the 3-wire serial I/O mode is as follows.

<1> Set the CSIC10 register (see **Figures 17-3**).
<2> Set bits 0, 4, and 6 (CSOT10, DIR10, and TRMD10) of the CSIM10 register (see **Figures 17-2**).
<3> Set bit 7 (CSIE10) of the CSIM10 register to 1. → Transmission/reception is enabled.
<4> Write data to transmit buffer register 10 (SOTB10). → Data transmission/reception is started.
Read data from serial I/O shift register 10 (SIO10). → Data reception is started.

Caution Take relationship with the other party of communication when setting the port mode register and port register.

The relationship between the register settings and pins is shown below.

Table 17-2. Relationship Between Register Settings and Pins

CSIE10	TRMD10	PM16	P16	PM17	P17	PM15	P15	CSI10 Operation	Pin Function		
									SI10/P16	SO10/P17 /FLMD1	$\overline{\text{SCK10}}$ / P15
0	×	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	×[Note 1]	Stop	P16	P17 /FLMD1	P15[Note 2]
1	0	1	×	×[Note 1]	×[Note 1]	1	×	Slave reception[Note 3]	SI10	P17 /FLMD1	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	×[Note 1]	×[Note 1]	0	0	1	×	Slave transmission[Note 3]	P16	SO10	$\overline{\text{SCK10}}$ (input)[Note 3]
1	1	1	×	0	0	1	×	Slave transmission/ reception[Note 3]	SI10	SO10	$\overline{\text{SCK10}}$ (input)[Note 3]
1	0	1	×	×[Note 1]	×[Note 1]	0	1	Master reception	SI10	P17 /FLMD1	$\overline{\text{SCK10}}$ (output)
1	1	×[Note 1]	×[Note 1]	0	0	0	1	Master transmission	P16	SO10	$\overline{\text{SCK10}}$ (output)
1	1	1	×	0	0	0	1	Master transmission/ reception	SI10	SO10	$\overline{\text{SCK10}}$ (output)

Notes 1. Can be set as port function.

2. To use P15/$\overline{\text{SCK10}}$ as port pins, clear CKP10 to 0.

3. To use the slave mode, set CKS102, CKS101, and CKS100 to 1, 1, 1.

Remark	
×:	don't care
CSIE10:	Bit 7 of serial operation mode register 10 (CSIM10)
TRMD10:	Bit 6 of CSIM10
CKP10:	Bit 4 of serial clock selection register 10 (CSIC10)
CKS102, CKS101, CKS100:	Bits 2 to 0 of CSIC10
PM1×:	Port mode register
P1×:	Port output latch

(2) Communication operation

In the 3-wire serial I/O mode, data is transmitted or received in 8-bit units. Each bit of the data is transmitted or received in synchronization with the serial clock.

Data can be transmitted or received if bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 1. Transmission/reception is started when a value is written to transmit buffer register 10 (SOTB10). In addition, data can be received when bit 6 (TRMD10) of serial operation mode register 10 (CSIM10) is 0.

Reception is started when data is read from serial I/O shift register 10 (SIO10).

After communication has been started, bit 0 (CSOT10) of CSIM10 is set to 1. When communication of 8-bit data has been completed, a communication completion interrupt request flag (CSIIF10) is set, and CSOT10 is cleared to 0. Then the next communication is enabled.

Caution Do not access the control register and data register when CSOT10 = 1 (during serial communication).

Figure 17-5. Timing in 3-Wire Serial I/O Mode (1/2)

(1) Transmission/reception timing (Type 1; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 0)

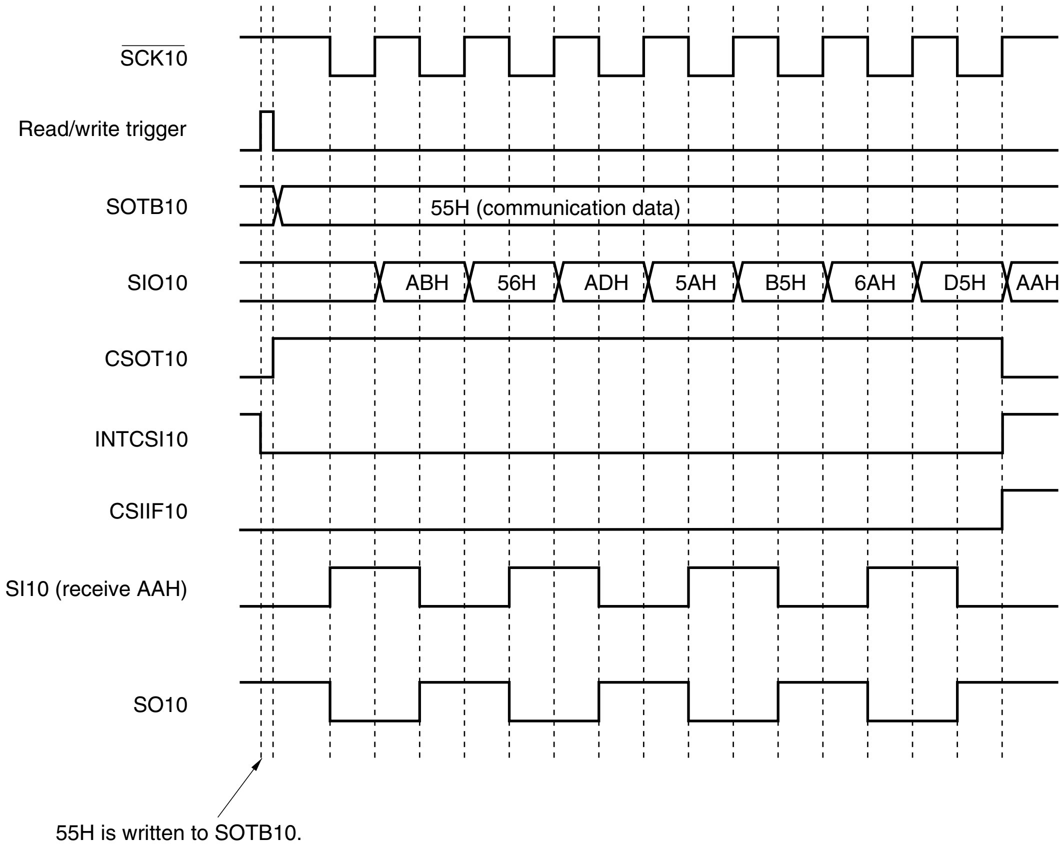


Figure 17-5. Timing in 3-Wire Serial I/O Mode (2/2)

(2) Transmission/reception timing (Type 2; TRMD10 = 1, DIR10 = 0, CKP10 = 0, DAP10 = 1)

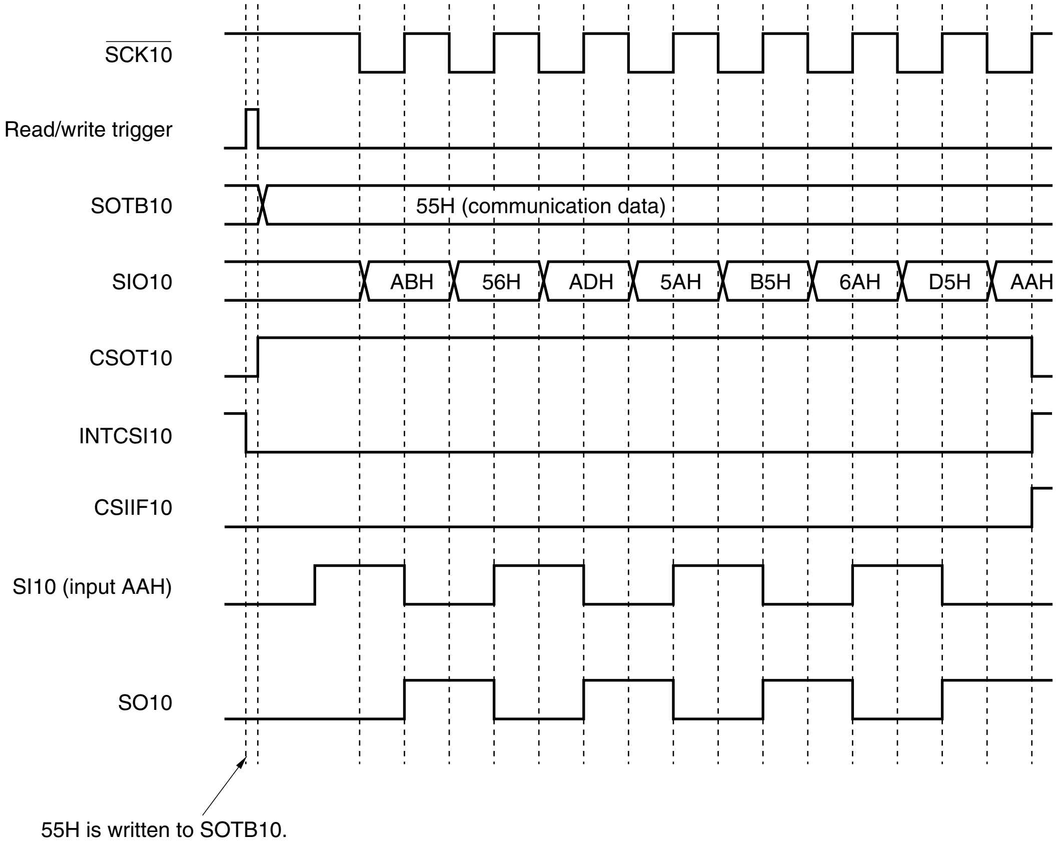
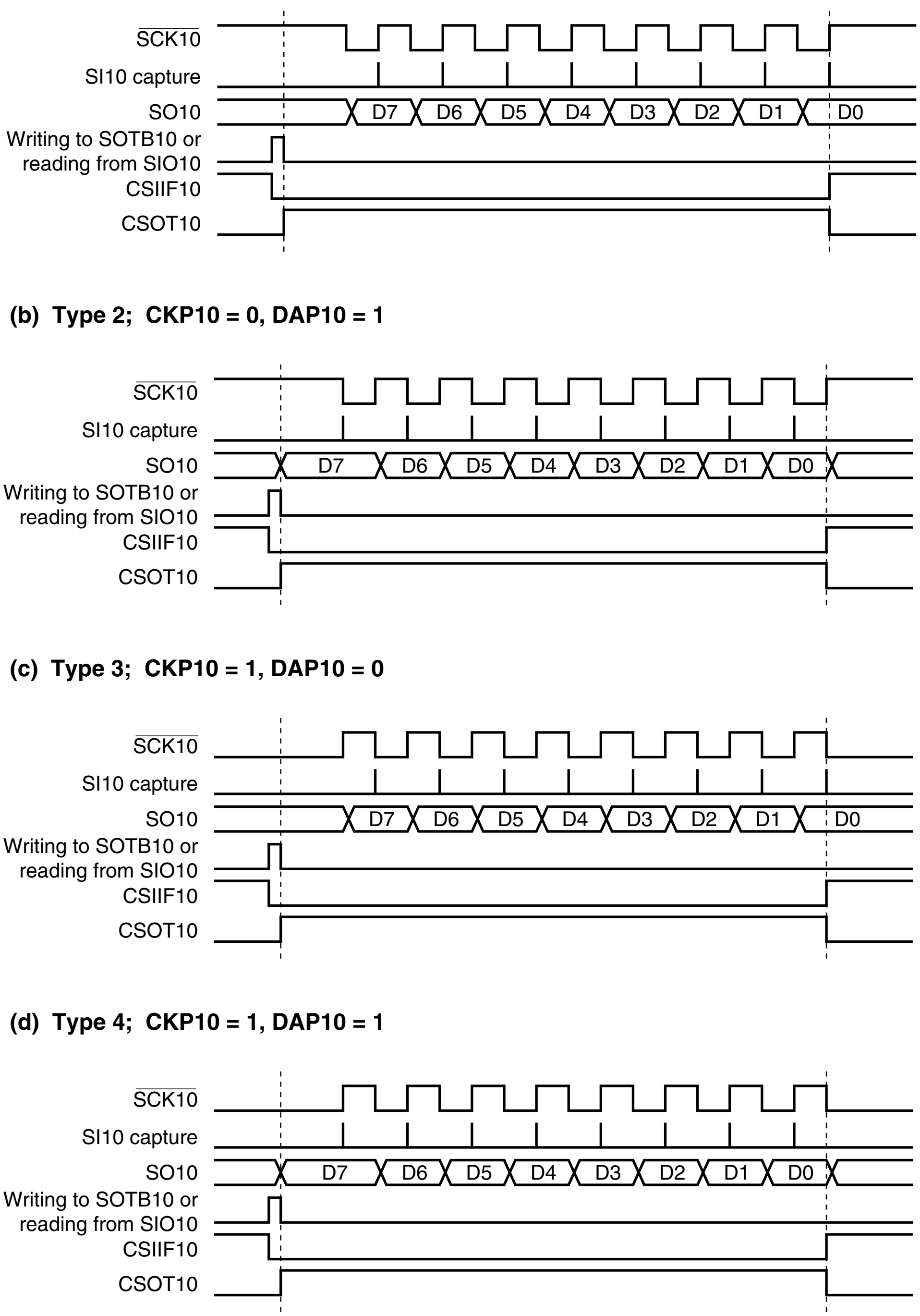


Figure 17-6. Timing of Clock/Data Phase



(a) Type 1; CKP10 = 0, DAP10 = 0

(b) Type 2; CKP10 = 0, DAP10 = 1

(c) Type 3; CKP10 = 1, DAP10 = 0

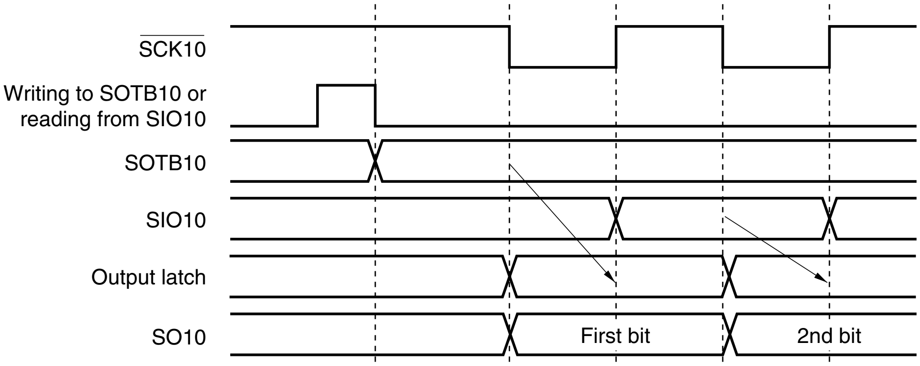
(d) Type 4; CKP10 = 1, DAP10 = 1

(3) Timing of output to SO10 pin (first bit)

When communication is started, the value of transmit buffer register 10 (SOTB10) is output from the SO10 pin. The output operation of the first bit at this time is described below.

Figure 17-7. Output Operation of First Bit

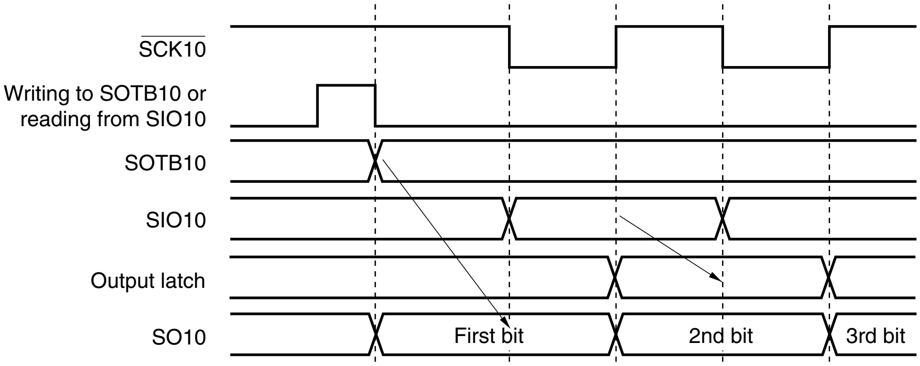
(1) When CKP10 = 0, DAP10 = 0 (or CKP10 = 1, DAP10 = 0)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
SOTB10
SIO10
Output latch
SO10
First bit
2nd bit

The first bit is directly latched by the SOTB10 register to the output latch at the falling (or rising) edge of $\overline{\text{SCK10}}$, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next rising (or falling) edge of $\overline{\text{SCK10}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

The second and subsequent bits are latched by the SIO10 register to the output latch at the next falling (or rising) edge of $\overline{\text{SCK10}}$, and the data is output from the SO10 pin.

(2) When CKP10 = 0, DAP10 = 1 (or CKP10 = 1, DAP10 = 1)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
SOTB10
SIO10
Output latch
SO10
First bit
2nd bit
3rd bit

The first bit is directly latched by the SOTB10 register at the falling edge of the write signal of the SOTB10 register or the read signal of the SIO10 register, and output from the SO10 pin via an output selector. Then, the value of the SOTB10 register is transferred to the SIO10 register at the next falling (or rising) edge of $\overline{\text{SCK10}}$, and shifted one bit. At the same time, the first bit of the receive data is stored in the SIO10 register via the SI10 pin.

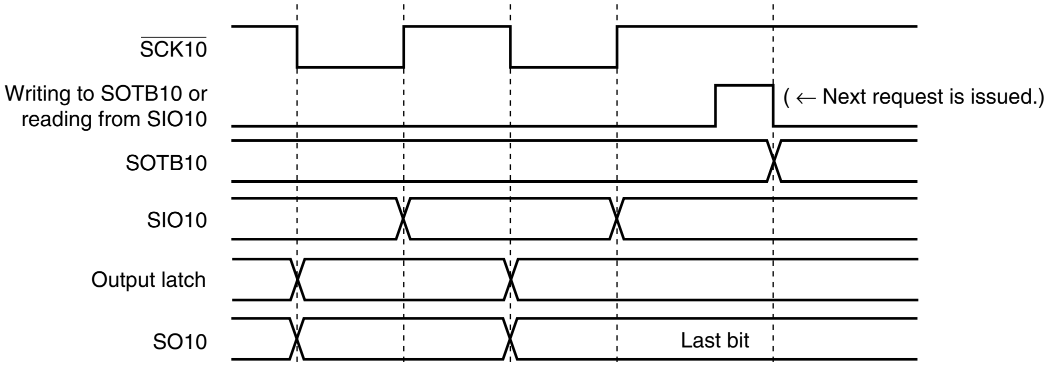
The second and subsequent bits are latched by the SIO10 register to the output latch at the next rising (or falling) edge of $\overline{\text{SCK10}}$, and the data is output from the SO10 pin.

(4) Output value of SO10 pin (last bit)

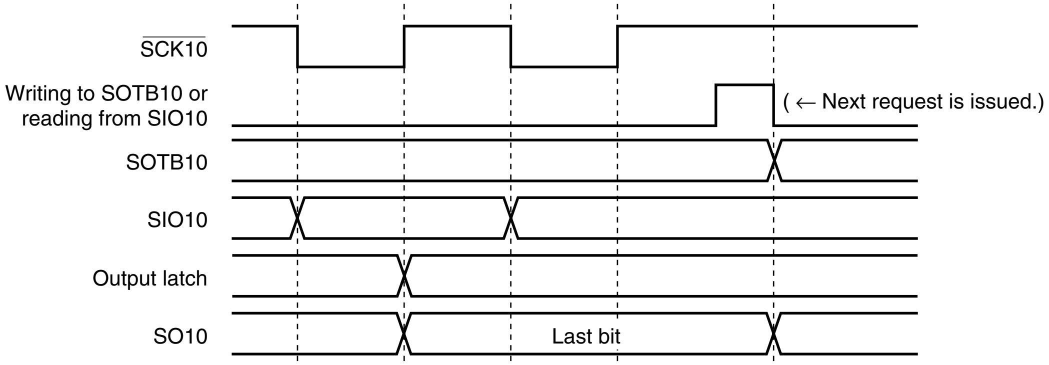
After communication has been completed, the SO10 pin holds the output value of the last bit.

Figure 17-8. Output Value of SO10 Pin (Last Bit)

(1) Type 1; when CKP10 = 0 and DAP10 = 0 (or CKP10 = 1, DAP10 = 0)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
(← Next request is issued.)
SOTB10
SIO10
Output latch
SO10
Last bit

(2) Type 2; when CKP10 = 0 and DAP10 = 1 (or CKP10 = 1, DAP10 = 1)


$\overline{\text{SCK10}}$
Writing to SOTB10 or reading from SIO10
(← Next request is issued.)
SOTB10
SIO10
Output latch
SO10
Last bit

(5) SO10 output

The status of the SO10 output is as follows if bit 7 (CSIE10) of serial operation mode register 10 (CSIM10) is cleared to 0.

Table 17-3. SO10 Output Status

TRMD10	DAP10	DIR10	SO10 Output Note 1
TRMD10 = 0 Note 2	–	–	Outputs low level Note 2
TRMD10 = 1	DAP10 = 0	–	Value of SO10 latch (low-level output)
	DAP10 = 1	DIR10 = 0	Value of bit 7 of SOTB10
		DIR10 = 1	Value of bit 0 of SOTB10

Notes 1. Actual output of SO10/P17/FLMD1 pin is decided by PM17 and P17 besides SO10 output.

2. Status after reset

Caution If a value is written to TRMD10, DAP10, and DIR10, the output value of SO10 changes.

CHAPTER 18 MULTIPLIER/DIVIDER

18.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 16 bits = 32 bits, 16-bit remainder (division)

18.2 Configuration of Multiplier/Divider

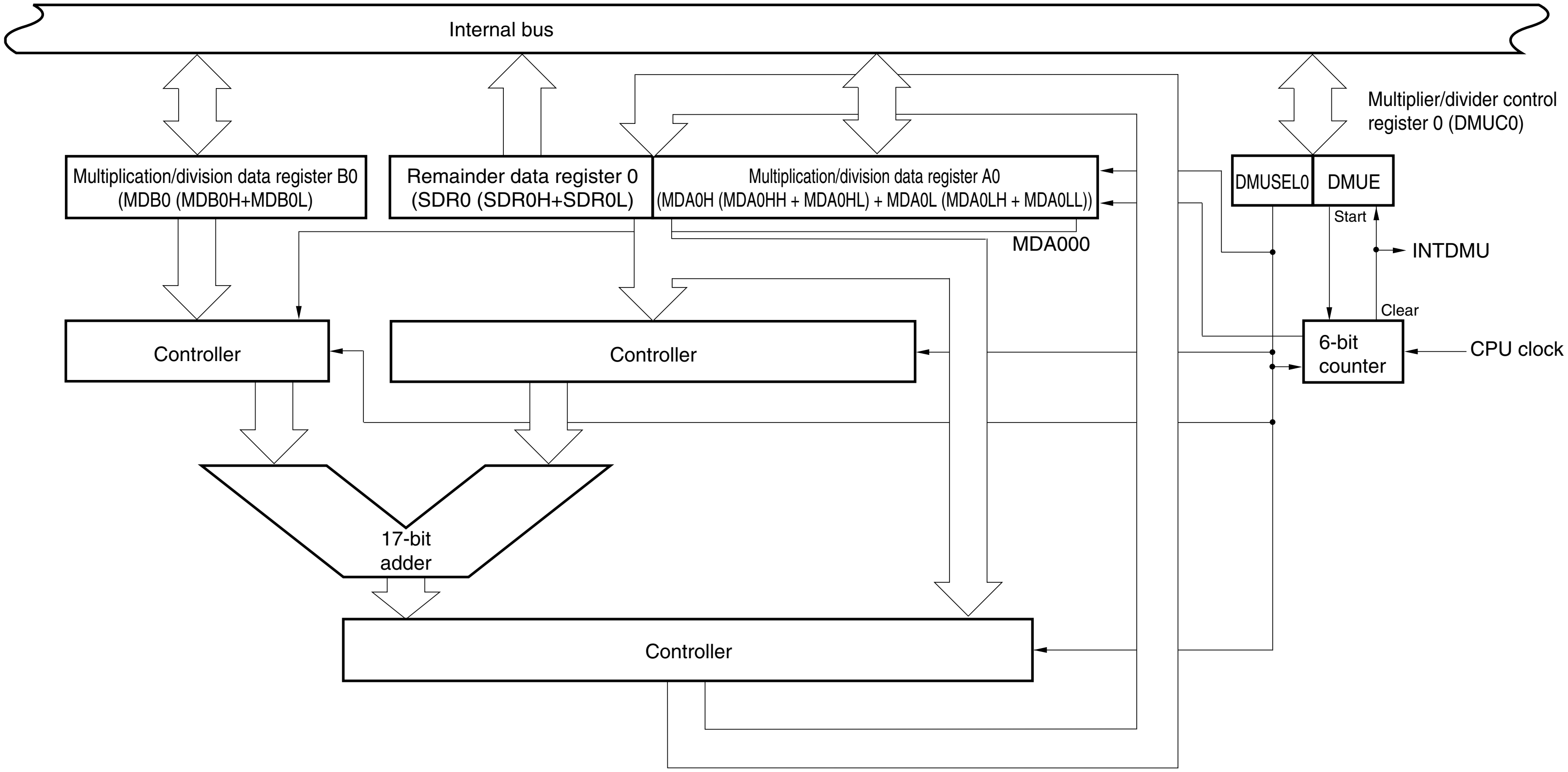
The multiplier/divider consists of the following hardware.

Table 18-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Remainder data register 0 (SDR0) Multiplication/division data registers A0 (MDA0H, MDA0L) Multiplication/division data registers B0 (MDB0)
Control register	Multiplier/divider control register 0 (DMUC0)

Figure 18-1 shows the block diagram of the multiplier/divider.

Figure 18-1. Block Diagram of Multiplier/Divider


Internal bus
Multiplier/divider control register 0 (DMUC0)
Multiplication/division data register B0 (MDB0 (MDB0H+MDB0L)
Remainder data register 0 (SDR0 (SDR0H+SDR0L)
Multiplication/division data register A0 (MDA0H (MDA0HH + MDA0HL) + MDA0L (MDA0LH + MDA0LL))
MDA000
DMUSEL0
DMUE
Start
INTDMU
Clear
6-bit counter
CPU clock
Controller
Controller
17-bit adder
Controller

(1) Remainder data register 0 (SDR0)

SDR0 is a 16-bit register that stores a remainder. This register stores 0 in the multiplication mode and the remainder of an operation result in the division mode.

This register can be read by an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 18-2. Format of Remainder Data Register 0 (SDR0)

Address: FF60H, FF61H After reset: 0000H R

Symbol	FF61H (SDR0H)								FF60H (SDR0L)							
SDR0	SDR 015	SDR 014	SDR 013	SDR 012	SDR 011	SDR 010	SDR 009	SDR 008	SDR 007	SDR 006	SDR 005	SDR 004	SDR 003	SDR 002	SDR 001	SDR 000

Cautions 1. The value read from SDR0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1) is not guaranteed.

2. SDR0 is reset when the operation is started (when DMUE is set to 1).

(2) Multiplication/division data register A0 (MDA0H, MDA0L)

MDA0 is a 32-bit register that sets a 16-bit multiplier A in the multiplication mode and a 32-bit dividend in the division mode, and stores the 32-bit result of the operation (higher 16 bits: MDA0H, lower 16 bits: MDA0L).

Figure 18-3. Format of Multiplication/Division Data Register A0 (MDA0H, MDA0L)

Address: FF62H, FF63H, FF64H, FF65H After reset: 0000H, 0000H R/W

Symbol	FF65H (MDA0HH)								FF64H (MDA0HL)							
MDA0H	MDA 031	MDA 030	MDA 029	MDA 028	MDA 027	MDA 026	MDA 025	MDA 024	MDA 023	MDA 022	MDA 021	MDA 020	MDA 019	MDA 018	MDA 017	MDA 016

Symbol	FF63H (MDA0LH)								FF62H (MDA0LL)							
MDA0L	MDA 015	MDA 014	MDA 013	MDA 012	MDA 011	MDA 010	MDA 009	MDA 008	MDA 007	MDA 006	MDA 005	MDA 004	MDA 003	MDA 002	MDA 001	MDA 000

Cautions 1. MDA0H is cleared to 0 when an operation is started in the multiplication mode (when multiplier/divider control register 0 (DMUC0) is set to 81H).

2. Do not change the value of MDA0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

3. The value read from MDA0 during operation processing (while DMUE is 1) is not guaranteed.

The functions of MDA0 when an operation is executed are shown in the table below.

Table 18-2. Functions of MDA0 During Operation Execution

DMUSEL0	Operation Mode	Setting	Operation Result
0	Division mode	Dividend	Division result (quotient)
1	Multiplication mode	Higher 16 bits: 0, Lower 16 bits: Multiplier A	Multiplication result (product)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

  <Multiplier A> <Multiplier B> <Product>
  MDA0 (bits 15 to 0) × MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0)

- Register configuration during division

  <Dividend> <Divisor> <Quotient> <Remainder>
  MDA0 (bits 31 to 0) ÷ MDB0 (bits 15 to 0) = MDA0 (bits 31 to 0) … SDR0 (bits 15 to 0)

MDA0 fetches the calculation result as soon as the clock is input, when bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is set to 1.

MDA0H and MDA0L can be set by an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 0000H.

(3) Multiplication/division data register B0 (MDB0)

MDB0 is a register that stores a 16-bit multiplier B in the multiplication mode and a 16-bit divisor in the division mode.

This register can be set by an 8-bit or 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 0000H.

Figure 18-4. Format of Multiplication/Division Data Register B0 (MDB0)

Address: FF66H, FF67H After reset: 0000H R/W

Symbol	FF67H (MDB0H)								FF66H (MDB0L)							
MDB0	MDB 015	MDB 014	MDB 013	MDB 012	MDB 011	MDB 010	MDB 009	MDB 008	MDB 007	MDB 006	MDB 005	MDB 004	MDB 003	MDB 002	MDB 001	MDB 000

Cautions 1. Do not change the value of MDB0 during operation processing (while bit 7 (DMUE) of multiplier/divider control register 0 (DMUC0) is 1). Even in this case, the operation is executed, but the result is undefined.

2. Do not clear MDB0 to 0000H in the division mode. If set, undefined operation results are stored in MDA0 and SDR0.

18.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by multiplier/divider control register 0 (DMUC0).

(1) Multiplier/divider control register 0 (DMUC0)

DMUC0 is an 8-bit register that controls the operation of the multiplier/divider.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 18-5. Format of Multiplier/Divider Control Register 0 (DMUC0)

Address: FF68H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
DMUC0	DMUE	0	0	0	0	0	0	DMUSEL0

DMUE[Note]	Operation start/stop
0	Stops operation
1	Starts operation

DMUSEL0	Operation mode (multiplication/division) selection
0	Division mode
1	Multiplication mode

Note When DMUE is set to 1, the operation is started. DMUE is automatically cleared to 0 after the operation is complete.

Cautions 1. If DMUE is cleared to 0 during operation processing (when DMUE is 1), the operation result is not guaranteed. If the operation is completed while the clearing instruction is being executed, the operation result is guaranteed, provided that the interrupt flag is set.

2. Do not change the value of DMUSEL0 during operation processing (while DMUE is 1). If it is changed, undefined operation results are stored in multiplication/division data register A0 (MDA0) and remainder data register 0 (SDR0).

3. If DMUE is cleared to 0 during operation processing (while DMUE is 1), the operation processing is stopped. To execute the operation again, set multiplication/division data register A0 (MDA0), multiplication/division data register B0 (MDB0), and multiplier/divider control register 0 (DMUC0), and start the operation (by setting DMUE to 1).

18.4 Operations of Multiplier/Divider

18.4.1 Multiplication operation

- Initial setting

1. Set operation data to multiplication/division data register A0L (MDA0L) and multiplication/division data register B0 (MDB0).
2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 1. Operation will start.

- During operation

3. The operation will be completed when 16 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers during operation, and therefore the read values of these registers are not guaranteed).

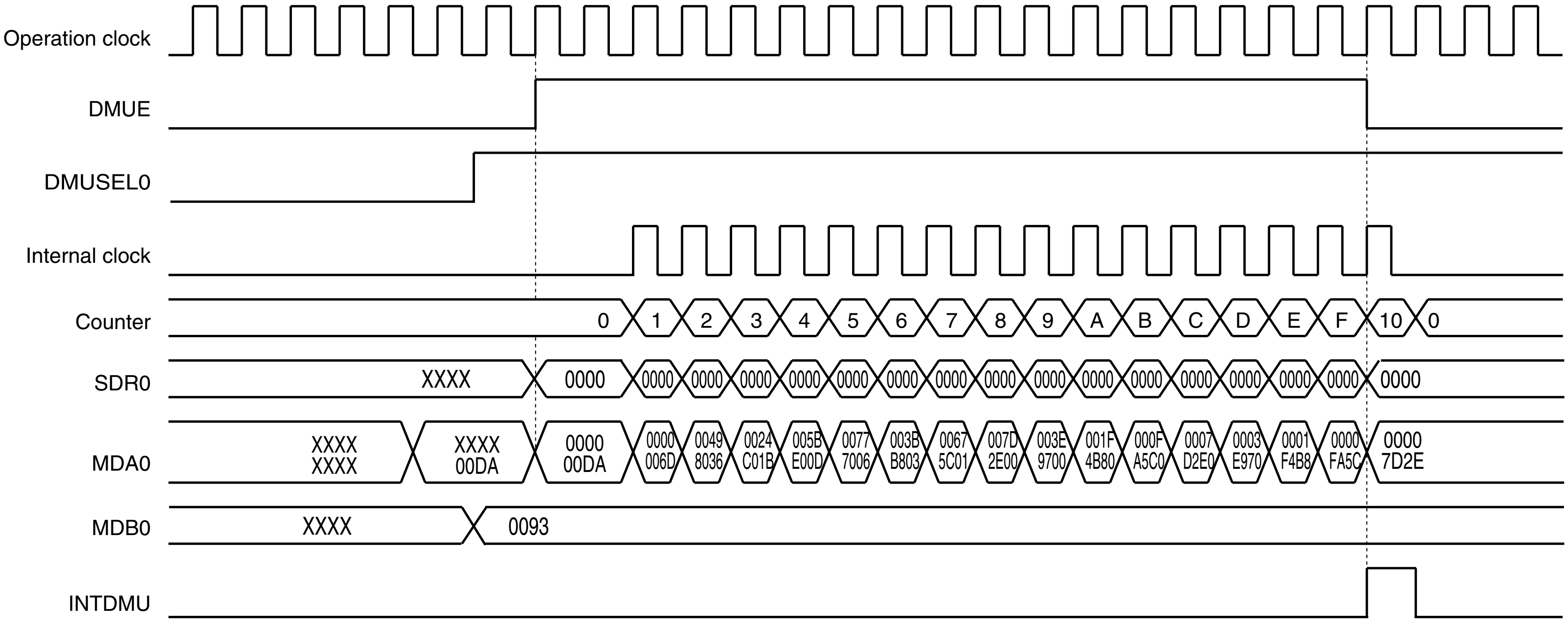
- End of operation

4. The operation result data is stored in the MDA0L and MDA0H registers.
5. DMUE is cleared to 0 (end of operation).
6. After the operation, an interrupt request signal (INTDMU) is generated.

- Next operation

7. To execute multiplication next, start from the initial setting in **18.4.1 Multiplication operation**.
8. To execute division next, start from the initial setting in **18.4.2 Division operation**.

Figure 18-6. Timing Chart of Multiplication Operation (00DAH × 0093H)



18.4.2 Division operation

- Initial setting

1. Set operation data to multiplication/division data register A0 (MDA0L and MDA0H) and multiplication/division data register B0 (MDB0).
2. Set bits 0 (DMUSEL0) and 7 (DMUE) of multiplier/divider control register 0 (DMUC0) to 0 and 1, respectively. Operation will start.

- During operation

3. The operation will be completed when 32 internal clocks have been issued after the start of the operation (intermediate data is stored in the MDA0L and MDA0H registers and remainder data register 0 (SDR0) during operation, and therefore the read values of these registers are not guaranteed).

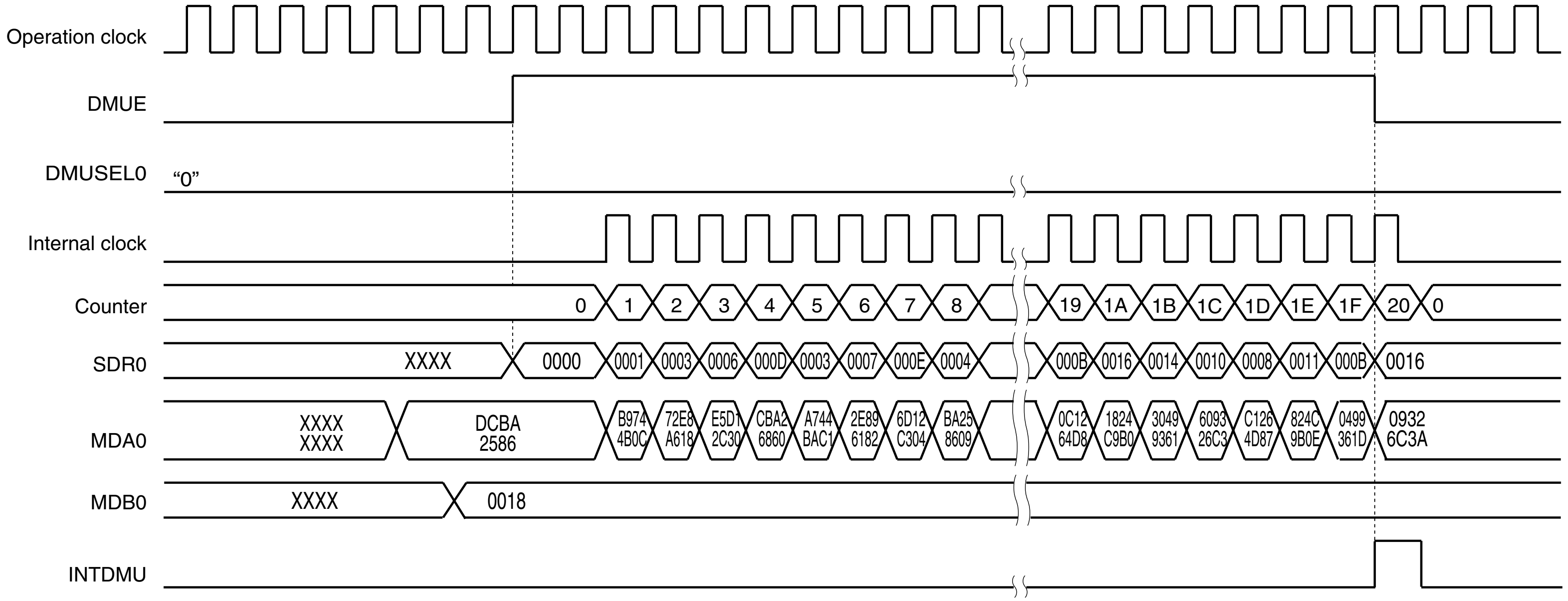
- End of operation

4. The result data is stored in the MDA0L, MDA0H, and SDR0 registers.
5. DMUE is cleared to 0 (end of operation).
6. After the operation, an interrupt request signal (INTDMU) is generated.

- Next operation

7. To execute multiplication next, start from the initial setting in **18.4.1 Multiplication operation**.
8. To execute division next, start from the initial setting in **18.4.2 Division operation**.

Figure 18-7. Timing Chart of Division Operation (DCBA2586H ÷ 0018H)



CHAPTER 19 INTERRUPT FUNCTIONS

19.1 Interrupt Function Types

The following three types of interrupt functions are used.

<R> **(1) Non-maskable interrupt**

A non-maskable interrupt is acknowledged even when interrupts are disabled. It does not undergo priority control and is given top priority over all other interrupt requests. However, interrupt requests are held pending during non-maskable interrupt servicing.

A non-maskable interrupt generates a standby release signal and releases the STOP mode and HALT mode.

The only non-maskable interrupt in the μPD78F0714 is the interrupt from the low-voltage detector.

(2) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into a high interrupt priority group and a low interrupt priority group by setting the priority specification flag registers (PR0L, PR0H, PR1L, PR1H).

Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupts with the same priority are generated simultaneously, each interrupt is serviced according to its predetermined priority (see **Table 19-1**).

A standby release signal is generated and STOP and HALT modes are released.

Eight external interrupt requests and 19 internal interrupt requests are provided as maskable interrupts.

(3) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

19.2 Interrupt Sources and Configuration

A total of 29 interrupt sources exist among non-maskable, maskable, and software interrupts (see **Table 19-1**).

Table 19-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
<R> Non-maskable	–	INTLVI	Low-voltage detection [Note 3]	Internal	0004H	(A)
Maskable	0	INTP0	Pin input edge detection	External	0006H	(B)
	1	INTP1			0008H	
	2	INTP2			000AH	
	3	INTP3			000CH	
	4	INTP4			000EH	
	5	INTP5			0010H	
	6	INTP6			0012H	
	7	INTP7			0014H	
	8	INTTW0UD	TW0UDC underflow	Internal	0016H	(A)
	9	INTTW0CM3	Match between TW0UDC and TW0CM3		0018H	
	10	INTTW0CM4	Match between TW0UDC and TW0CM4		001AH	
	11	INTTW0CM5	Match between TW0UDC and TW0CM5		001CH	
	12	INTCM10	Match between IT20UDC and IT20CM0		001EH	
	13	INTCM11	Match between IT20UDC and IT20CM1		0020H	
	14	INTCC10	Match between IT20UDC and IT20CC0 (when compare register is specified), TIT20CC0 pin valid edge detection (when capture register is specified)		0022H	
	15	INTCC11	Match between IT20UDC and IT20CC1 (when compare register is specified), TIT20CC1 pin valid edge detection (when capture register is specified)		0024H	
	–	–	–		0026H [Note 4]	
	16	INTTM00	Match between TM00 and CR00 (when compare register is specified), TI001 pin valid edge detection (when capture register is specified)		0028H	
	17	INTTM01	Match between TM00 and CR01 (when compare register is specified), TI000 pin valid edge detection (when capture register is specified)		002AH	
	18	INTSRE00	UART00 reception error occurrence		002CH	
	19	INTSR00	End of UART00 reception		002EH	
	20	INTST00	End of UART00 transmission		0030H	

Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 26 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 19-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

4. There is no interrupt request corresponding to vector table address 0026H.

Table 19-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/ External	Vector Table Address	Basic Configuration Type[Note 2]
		Name	Trigger			
Maskable	21	INTTM50	Match between TM50 and CR50 (when compare register is specified)	Internal	0032H	(A)
	22	INTTM51	Match between TM51 and CR51 (when compare register is specified)		0034H	
	23	INTTMH0	Match between TMH0 and CMP00 (when compare register is specified)		0036H	
	24	INTCSI10	End of CSI10 communication		0038H	
	25	INTDMU	End of multiply/divide operation		003AH	
	26	INTAD	End of A/D conversion		003CH	
Software	–	BRK	BRK instruction execution	–	003EH	(C)
Reset	–	RESET	Reset input	–	0000H	–
		POC	Power-on-clear			
		LVI	Low-voltage detection [Note 3]			
		WDT	WDT overflow			

Notes 1. The default priority is the priority applicable when two or more maskable interrupts are generated simultaneously. 0 is the highest priority, and 26 is the lowest.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 19-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

Figure 19-1. Basic Configuration of Interrupt Function (1/2)

<R> **(A) Internal non-maskable interrupt**

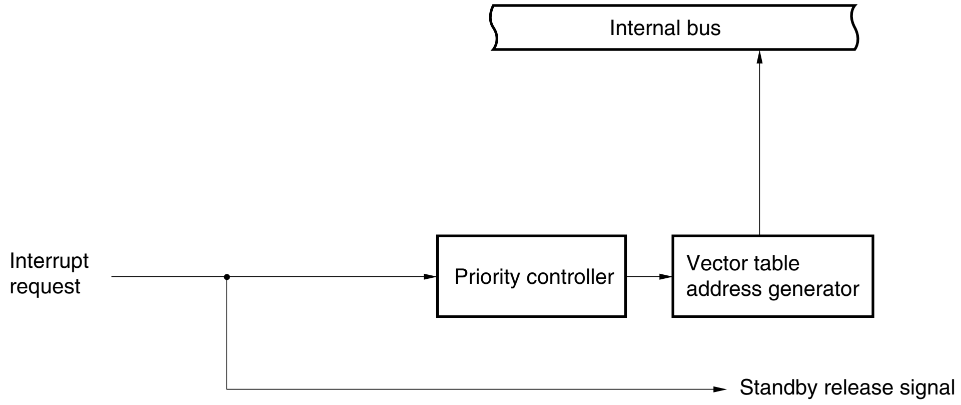
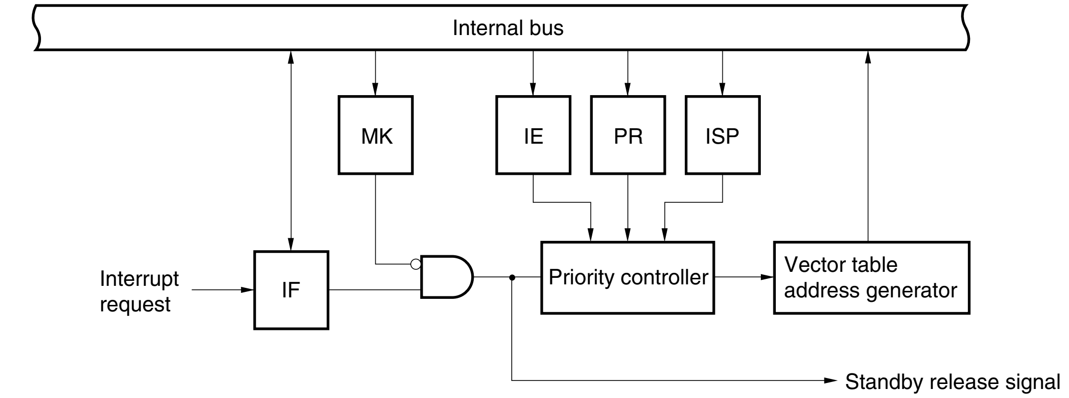
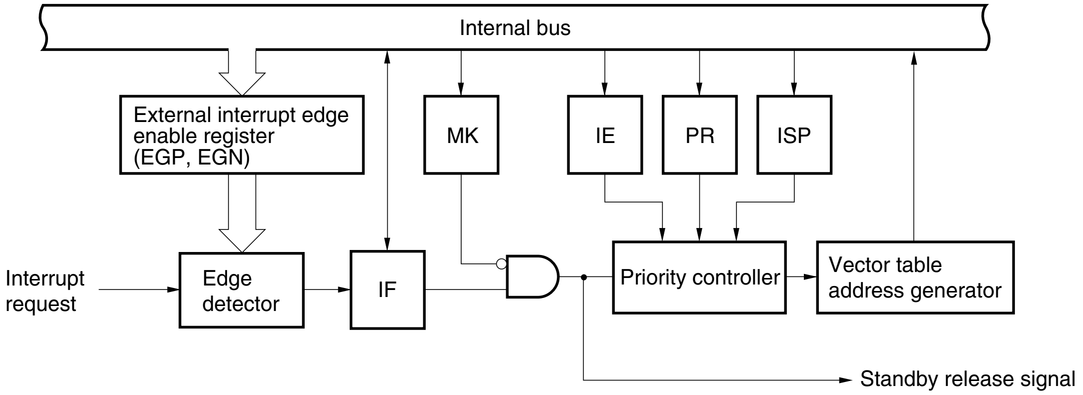


Figure 19-1. Basic Configuration of Interrupt Function (2/2)

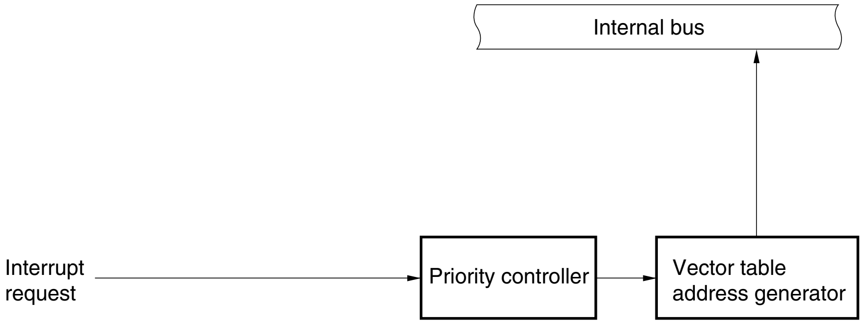
(B) Internal maskable interrupt



(C) External maskable interrupt (INTP0 to INTP7)



(D) Software interrupt



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP:	In-service priority flag
MK:	Interrupt mask flag
PR:	Priority specification flag

19.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag register (IF0L, IF0H, IF1L, IF1H)
- Interrupt mask flag register (MK0L, MK0H, MK1L, MK1H)
- Priority specification flag register (PR0L, PR0H, PR1L, PR1H)
- External interrupt rising edge enable register (EGP)
- External interrupt falling edge enable register (EGN)
- Program status word (PSW)

Table 19-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 19-2. Flags Corresponding to Interrupt Request Sources

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		Register		Register		Register
INTP0	PIF0	IF0L	PMK0	MK0L	PPR0	PR0L
INTP1	PIF1		PMK1		PPR1	
INTP2	PIF2		PMK2		PPR2	
INTP3	PIF3		PMK3		PPR3	
INTP4	PIF4		PMK4		PPR4	
INTP5	PIF5		PMK5		PPR5	
INTP6	PIF6		PMK6		PPR6	
INTP7	PIF7	IF0H	PMK7	MK0H	PPR7	PR0H
INTTW0UD	UDIFW0		UDMKW0		UDPRW0	
INTTW0CM3	CM3IFW0		CM3MKW0		CM3PRW0	
INTTW0CM4	CM4IFW0		CM4MKW0		CM4PRW0	
INTTW0CM5	CM5IFW0		CM5MKW0		CM5PRW0	
INTCM10	CMIF10		CMMK10		CMPR10	
INTCM11	CMIF11		CMMK11		CMPR11	
INTCC10	CCIF10		CCMK10		CCPR10	
INTCC11	CCIF11	IF1L	CCMK11	MK1L	CCPR11	PR1L
INTTM00	TMIF00		TMMK00		TMPR00	
INTTM01	TMIF01		TMMK01		TMPR01	
INTSRE00	SREIF00		SREMK00		SREPR00	
INTSR00	SRIF00		SRMK00		SRPR00	
INTST00	STIF00		STMK00		STPR00	
INTTM50	TMIF50		TMMK50		TMPR50	
INTTM51	TMIF51	IF1H	TMMK51	MK1H	TMPR51	PR1H
INTTMH0	TMIFH0		TMMKH0		TMPRH0	
INTCSI10	CSIIF10		CSIMK10		CSIPR10	
INTDMU	DMUIF		DMUMK		DMUPR	
INTAD	ADIF		ADMK		ADPR	

(1) Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon $\overline{\text{RESET}}$ input.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

IF0L, IF0H, IF1L, and IF1H are set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, and IF1L and IF1H are combined to form 16-bit registers IF0 and IF1, they are set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 19-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)

Address: FFE0H After reset: 00H R/W

<R>

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
IF0L	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	0[Note]

Address: FFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	CCIF10	CMIF11	CMIF10	CM5IFW0	CM4IFW0	CM3IFW0	UDIFW0	PIF7

Address: FFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
IF1L	TMIF50	STIF00	SRIF00	SREIF00	TMIF01	TMIF00	0[Note]	CCIF11

Address: FFE3H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
IF1H	0[Note]	0[Note]	0[Note]	ADIF	DMUIF	CSIIF10	TMIFH0	TMIF51

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Note Be sure to clear bit 0 of IF0L, bit 1 of IF1L, and bits 5 to 7 of IF1H to 0.

Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as “IF0L.0 = 0;” or “_asm(“clr1 IF0L, 0”);” because the compiled assemblermust be a 1-bit memory manipulation instruction (CLR1). If a program is described in C language using an 8-bit memory manipulation instruction such as “IF0L &= 0xfe;” and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between “mov a, IF0L” and “mov IF0L, a”, the flag is cleared to 0 at “mov IF0L, a”. Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, and MK1H are set by a 1-bit or 8-bit memory manipulation instruction. When MK0L and MK0H, and MK1L and MK1H are combined to form 16-bit registers MK0 and MK1, they are set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets MK0L, MK0H, and MK1L to FFH and MK1H to DFH.

Figure 19-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)

Address: FFE4H After reset: FFH R/W

<R>

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
MK0L	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	1[Note]

Address: FFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	CCMK10	CMMK11	CMMK10	CM5MKW0	CM4MKW0	CM3MKW0	UDMKW0	PMK7

Address: FFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
MK1L	TMMK50	STMK00	SRMK00	SREMK00	TMMK01	TMMK00	1[Note]	CCMK11

Address: FFE7H After reset: DFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
MK1H	1[Note]	1[Note]	0[Note]	ADMK	DMUMK	CSIMK10	TMMKH0	TMMK51

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note Be sure to set bit 0 of MK0L, bit 1 of MK1L, and bits 6 and 7 of MK1H to 1.
Be sure to clear bit 5 of MK1H to 0.

(3) Priority specification flag registers (PR0L, PR0H, PR1L, PR1H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority order.

PR0L, PR0H, PR1L, and PR1H are set by a 1-bit or 8-bit memory manipulation instruction. If PR0L and PR0H, and PR1L and PR1H are combined to form 16-bit registers PR0 and PR1, they are set by a 16-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets these registers to FFH.

Figure 19-4. Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)

Address: FFE8H After reset: FFH R/W

<R>

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	0
PR0L	PPR6	PPR5	PPR4	PPR3	PPR2	PPR1	PPR0	1[Note]

Address: FFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR0H	CCPR10	CMPR11	CMPR10	CM5PRW0	CM4PRW0	CM3PRW0	UDPRW0	PPR7

Address: FFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
PR1L	TMPR50	STPR00	SRPR00	SREPR00	TMPR01	TMPR00	1[Note]	CCPR11

Address: FFEBH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR1H	1[Note]	1[Note]	1[Note]	ADPR	DMUPR	CSIPR10	TMPRH0	TMPR51

XXPRX	Priority level selection
0	High priority level
1	Low priority level

Note Be sure to set bit 0 of PR0L, bit 1 of PR1L, and bits 5 to 7 of PR1H to 1.

(4) External interrupt rising edge enable register (EGP), external interrupt falling edge enable register (EGN)

These registers specify the valid edge for INTP0 to INTP7.

EGP and EGN are set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input clears these registers to 00H.

Figure 19-5. Format of External Interrupt Rising Edge Enable Register (EGP) and External Interrupt Falling Edge Enable Register (EGN)

Address: FF48H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP	EGP7	EPG6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FF49H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 19-3 shows the ports corresponding to EGPn and EGNn.

Table 19-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P00	INTP0
EGP1	EGN1	P01	INTP1
EGP2	EGN2	P02	INTP2
EGP3	EGN3	P03	INTP3
EGP4	EGN4	P52	INTP4
EGP5	EGN5	P53	INTP5
EGP6	EGN6	P55	INTP6
EGP7	EGN7	P56	INTP7

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 7

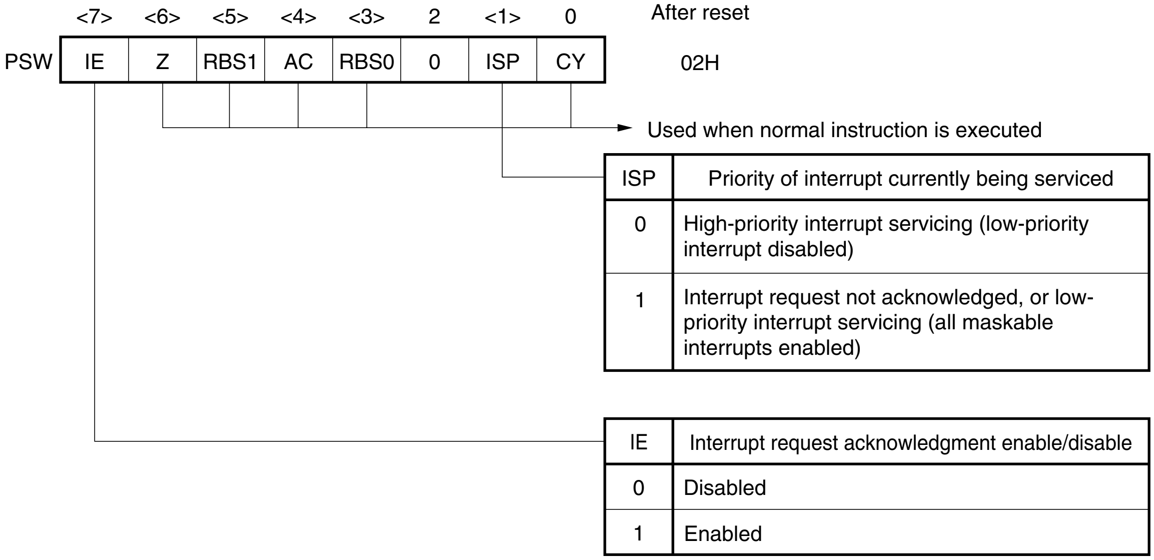
(5) Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP flag that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are transferred to the ISP flag. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

$\overline{\text{RESET}}$ input sets PSW to 02H.

Figure 19-6. Format of Program Status Word



	<7>	<6>	<5>	<4>	<3>	2	<1>	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	0	ISP	CY	02H

ISP	Priority of interrupt currently being serviced
0	High-priority interrupt servicing (low-priority interrupt disabled)
1	Interrupt request not acknowledged, or low-priority interrupt servicing (all maskable interrupts enabled)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

19.4 Interrupt Servicing Operations

<R> 19.4.1 Non-maskable interrupt request acknowledgment operation

A non-maskable interrupt request is unconditionally acknowledged even in an interrupt acknowledgment disabled state. It does not undergo interrupt priority control and has the highest priority of all interrupts.

If a non-maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag and ISP flag are reset (0), and the contents of the vector table are loaded into the PC and branched.

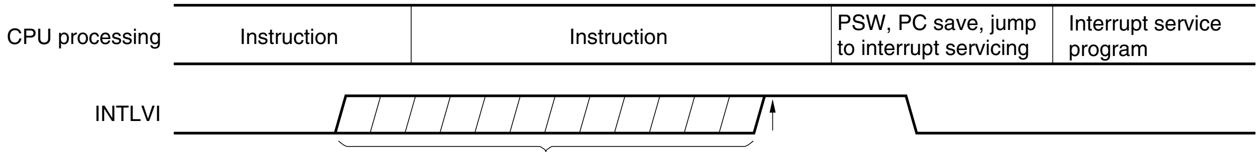
This disables the acknowledgment of multiple interrupts.

A new non-maskable interrupt request generated during execution of a non-maskable interrupt servicing program is acknowledged after the current non-maskable interrupt servicing program is terminated (following RETI instruction execution) and one main routine instruction has been executed. However, if a new non-maskable interrupt request is generated twice or more during non-maskable interrupt servicing program execution, only one non-maskable interrupt request is acknowledged after termination of the non-maskable interrupt servicing program.

Figures 19-7 and 19-8 show the acknowledgment timing of a non-maskable interrupt request, and the acknowledgment operation when multiple non-maskable interrupt requests are generated, respectively.

Caution Be sure to use the RETI instruction to restore processing from the non-maskable interrupt.

Figure 19-7. Non-Maskable Interrupt Request Acknowledgment Timing

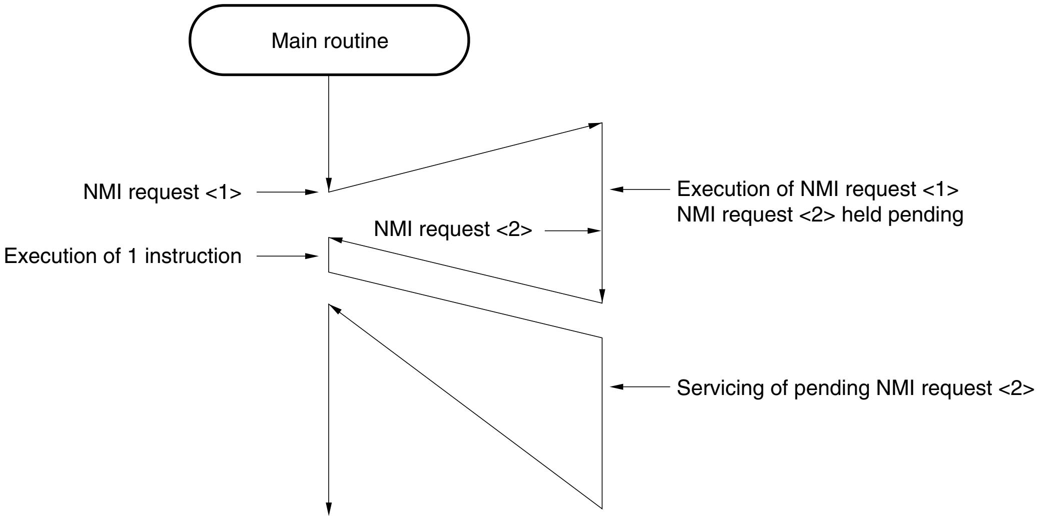

CPU processing
Instruction
Instruction
PSW, PC save, jump to interrupt servicing
Interrupt service program
INTLVI

Interrupt request generated during this interval is acknowledged at ↑.

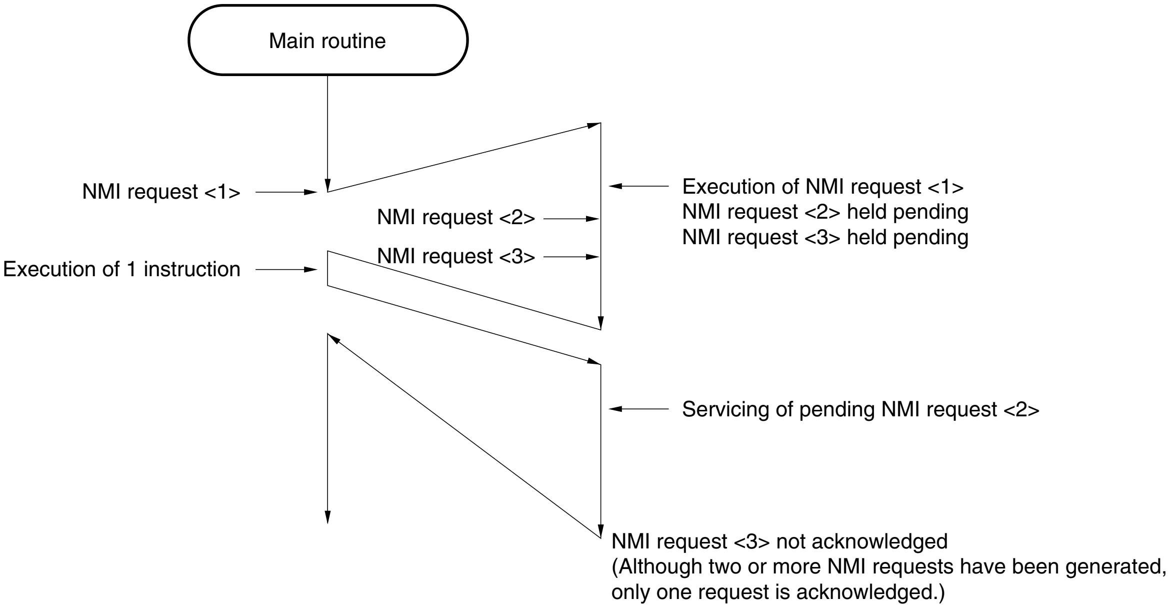
INTLVI: Low-voltage detector interrupt request signal

Figure 19-8. Non-Maskable Interrupt Request Acknowledgment Operation

(a) If a non-maskable interrupt request is generated during non-maskable interrupt servicing program execution


Main routine
NMI request <1>
Execution of NMI request <1>
NMI request <2> held pending
NMI request <2>
Execution of 1 instruction
Servicing of pending NMI request <2>

(b) If two non-maskable interrupt requests are generated during non-maskable interrupt servicing program execution


Main routine
NMI request <1>
Execution of NMI request <1>
NMI request <2> held pending
NMI request <3> held pending
NMI request <2>
NMI request <3>
Execution of 1 instruction
Servicing of pending NMI request <2>
NMI request <3> not acknowledged
(Although two or more NMI requests have been generated,
only one request is acknowledged.)

19.4.2 Maskable interrupt request acknowledgement

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request (when the ISP flag is reset to 0).

Moreover, even if the EI instruction is executed during execution of a non-maskable interrupt servicing program, neither non-maskable interrupt requests nor maskable interrupt requests are acknowledged.

The times from generation of a maskable interrupt request until interrupt servicing is performed are listed in Table 19-4 below.

For the interrupt request acknowledgment timing, see **Figures 19-10** and **19-11**.

Table 19-4. Time from Generation of Maskable Interrupt Request Until Servicing

	Minimum Time	Maximum Time[Note]
When ××PR = 0	7 clocks	32 clocks
When ××PR = 1	8 clocks	33 clocks

Note If an interrupt request is generated just before a divide instruction, the wait time becomes longer.

Remark 1 clock: 1/f_{CPU} (f_{CPU}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupt requests have the same priority level, the request with the highest default priority is acknowledged first.

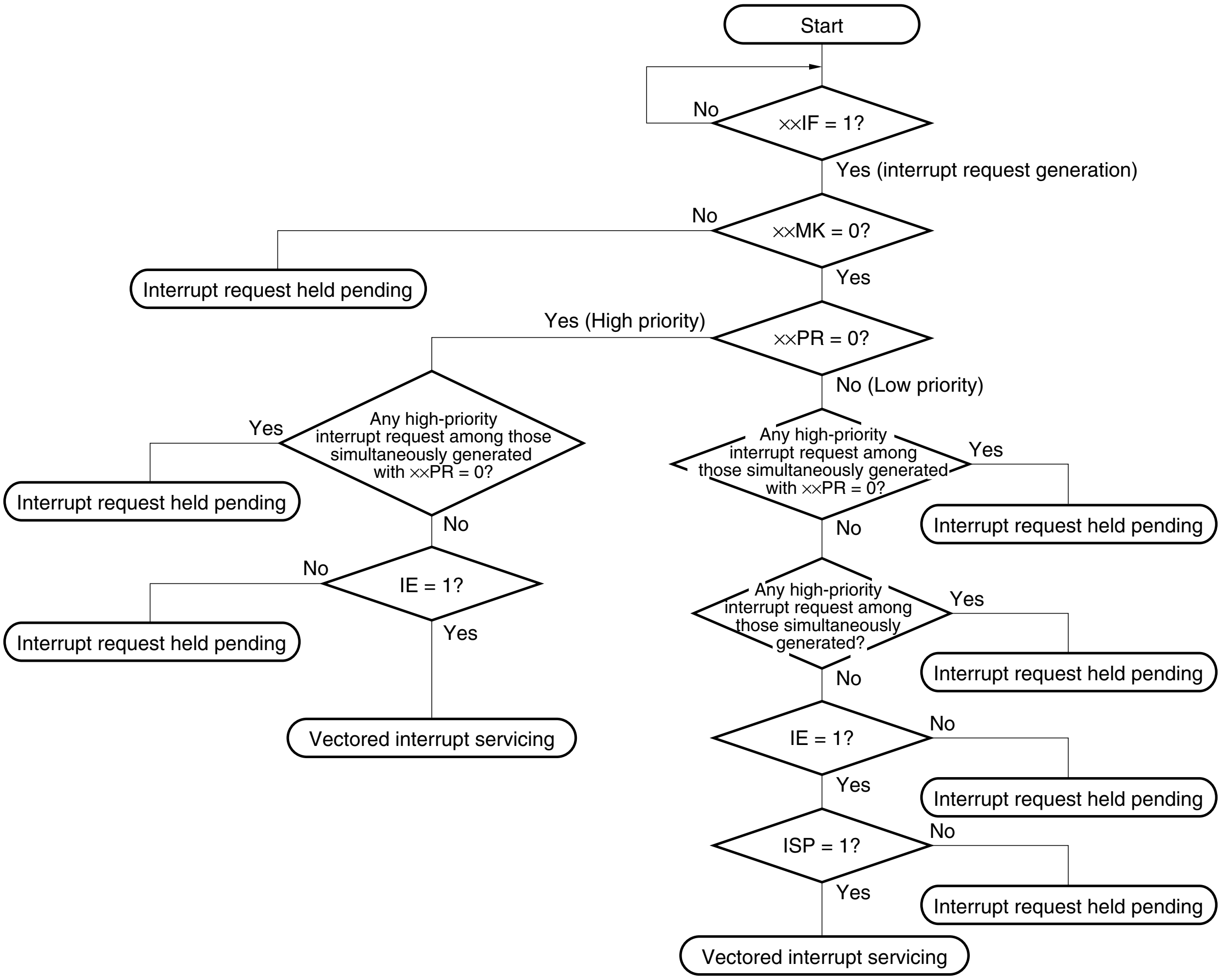
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 19-9 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP flag. The vector table data determined for each interrupt request is loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

Figure 19-9. Interrupt Request Acknowledgment Processing Algorithm


Start
No
××IF = 1?
Yes (interrupt request generation)
No
××MK = 0?
Interrupt request held pending
Yes
Yes (High priority)
××PR = 0?
No (Low priority)
Yes
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Interrupt request held pending
No
No
IE = 1?
Interrupt request held pending
Yes
Vectored interrupt servicing
Any high-priority interrupt request among those simultaneously generated with ××PR = 0?
Yes
Interrupt request held pending
No
Any high-priority interrupt request among those simultaneously generated?
Yes
Interrupt request held pending
No
IE = 1?
No
Yes
Interrupt request held pending
ISP = 1?
No
Yes
Interrupt request held pending
Vectored interrupt servicing

××IF: Interrupt request flag

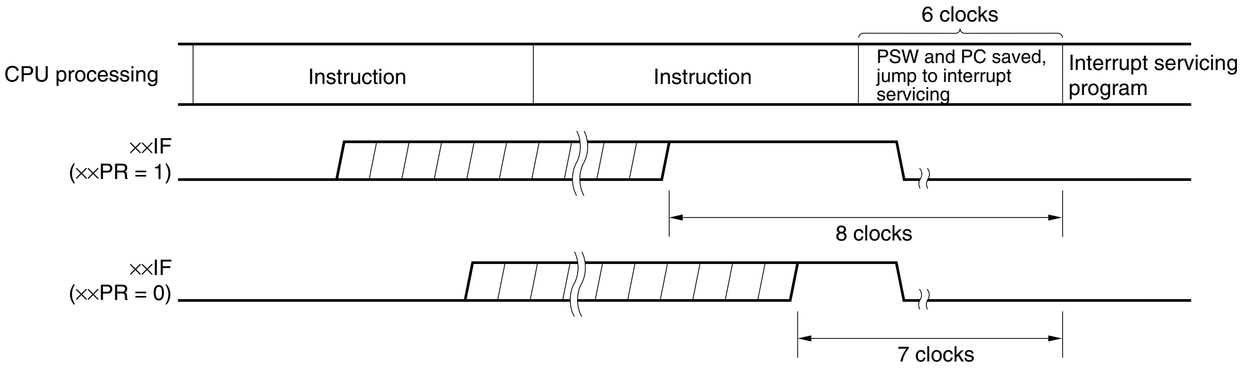
××MK: Interrupt mask flag

××PR: Priority specification flag

IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)

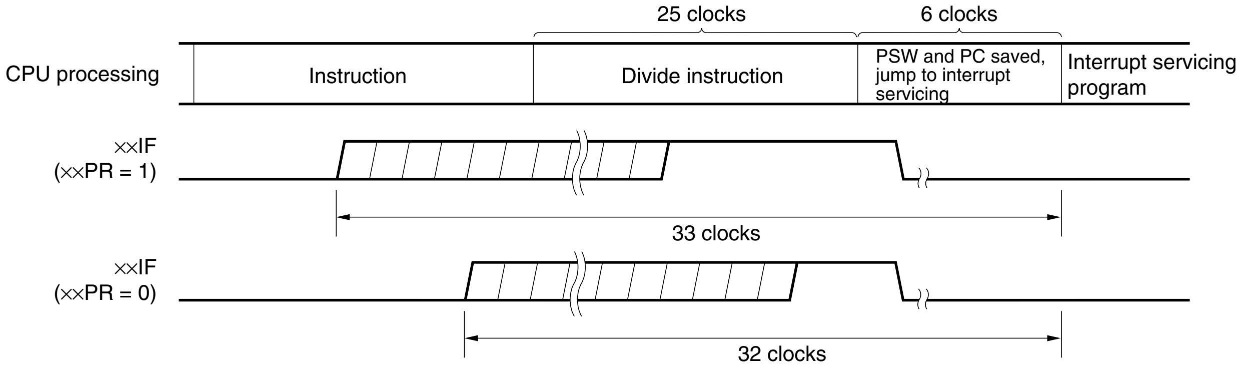
ISP: Flag that indicates the priority level of the interrupt currently being serviced (0 = high-priority interrupt servicing, 1 = No interrupt request acknowledged, or low-priority interrupt servicing)

Figure 19-10. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

Figure 19-11. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCPU (fCPU: CPU clock)

19.4.3 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (003EH, 003FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution Do not use the RETI instruction for restoring from the software interrupt.

19.4.4 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1) (except non-maskable interrupt). Also, when an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of one main processing instruction execution.

<R> Multiple interrupt servicing is not possible during non-maskable interrupt servicing.

Table 19-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 19-12 shows multiple interrupt servicing examples.

<R> **Table 19-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**

Multiple Interrupt Request Interrupt Being Serviced		Non-maskable Interrupt Request	Maskable Interrupt Request				Software Interrupt Request
			PR = 0		PR = 1		
			IE = 1	IE = 0	IE = 1	IE = 0	
Non-maskable interrupt request		×	×	×	×	×	○
Maskable interrupt	ISP = 0	○	○	×	×	×	○
	ISP = 1	○	○	×	○	×	○
Software interrupt		○	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

2. ×: Multiple interrupt servicing disabled

3. ISP and IE are flags contained in the PSW.

ISP = 0: An interrupt with higher priority is being serviced.

ISP = 1: No interrupt request has been acknowledged, or an interrupt with a lower priority is being serviced.

IE = 0: Interrupt request acknowledgment is disabled.

IE = 1: Interrupt request acknowledgment is enabled.

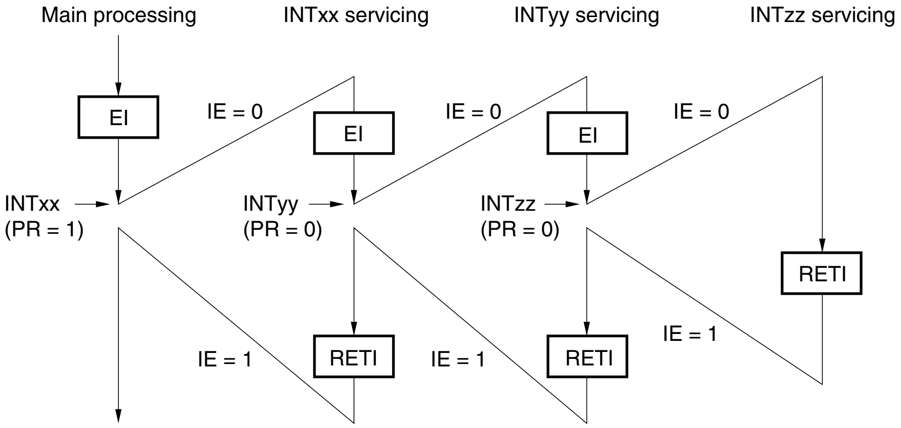
4. PR is a flag contained in PR0L, PR0H, PR1L, and PR1H.

PR = 0: Higher priority level

PR = 1: Lower priority level

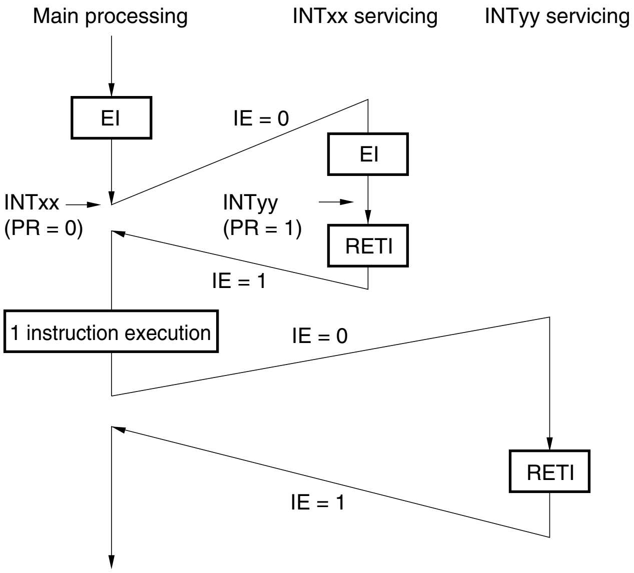
Figure 19-12. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

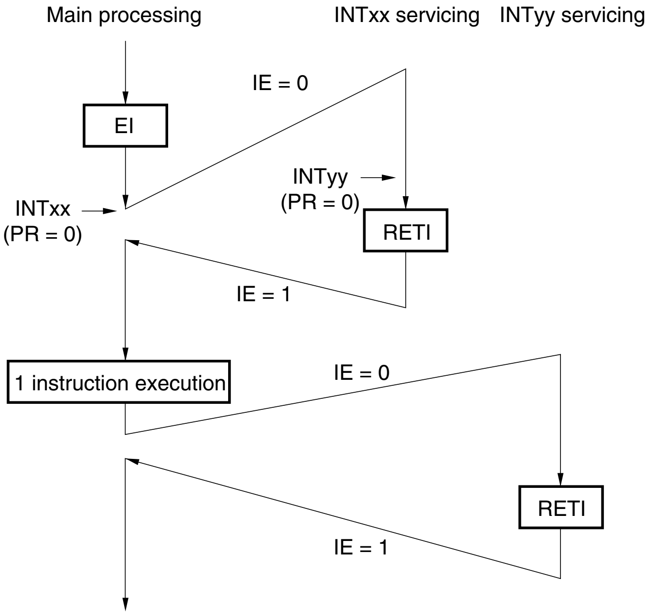


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
PR = 1: Lower priority level
IE = 0: Interrupt request acknowledgment disabled

Figure 19-12. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 0: Higher priority level
IE = 0: Interrupt request acknowledgment disabled

19.4.5 Interrupt request hold

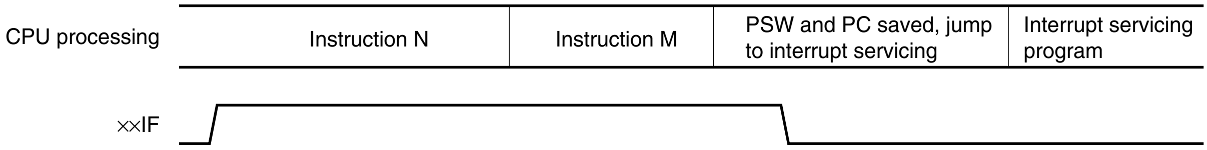
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV A, PSW
- MOV PSW, A
- MOV1 PSW.bit, CY
- MOV1 CY, PSW.bit
- AND1 CY, PSW.bit
- OR1 CY, PSW.bit
- XOR1 CY, PSW.bit
- SET1 PSW.bit
- CLR1 PSW.bit
- RETB
- RETI
- PUSH PSW
- POP PSW
- BT PSW.bit, $addr16
- BF PSW.bit, $addr16
- BTCLR PSW.bit, $addr16
- EI
- DI
- Manipulation instructions for the IF0L, IF0H, IF1L, IF1H, MK0L, MK0H, MK1L, MK1H, PR0L, PR0H, PR1L, and PR1H registers

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared to 0. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged. However, a non-maskable interrupt request is acknowledged.

Figure 19-13 shows the timing at which interrupt requests are held pending.

Figure 19-13. Interrupt Request Hold


CPU processing
Instruction N
Instruction M
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
××IF

Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 20 STANDBY FUNCTION

20.1 Standby Function and Configuration

20.1.1 Standby function

Table 20-1. Relationship Between Operation Clocks in Each Operation Status

Status / Operation Mode	X1 Oscillator		Internal Oscillator			CPU Clock After Release	Prescaler Clock Supplied to Peripherals	
	MSTOP = 0	MSTOP = 1	**Note 1**	**Note 2**			MCM0 = 0	MCM0 = 1
				RSTOP = 0	RSTOP = 1			
Reset	Stopped		Stopped			Internal oscillation clock	Stopped	
STOP			Oscillating	Oscillating	Stopped $^{\text{Note 4}}$	**Note 5**	Stopped	
HALT	Oscillating	Stopped $^{\text{Note 3}}$				**Note 6**	Internal oscillation clock	X1

Notes 1. When “Cannot be stopped” is selected for internal oscillator by an option byte.

2. When “Can be stopped by software” is selected for internal oscillator by an option byte.

3. Only when internal oscillator is oscillating.

4. Only when X1 is oscillating.

5. Operates using the CPU clock at STOP instruction execution.

6. Operates using the CPU clock at HALT instruction execution.

Caution The RSTOP setting is valid only when “Can be stopped by software” is set for internal oscillator by an option byte.

Remark MSTOP: Bit 7 of the main OSC control register (MOC)
RSTOP: Bit 0 of the internal oscillation mode register (RCM)
MCM0: Bit 0 of the main clock mode register (MCM)

The standby function is designed to reduce the operating current of the system. The following two modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the X1 oscillator or internal oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the X1 oscillator stops, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP and HALT modes can be used when the CPU operates with the X1 input clock or internal oscillation clock. However, when the STOP instruction is executed during internal oscillation clock operation, the X1 oscillator stops, but internal oscillator does not stop.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation before executing STOP instruction.

3. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) of the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the HALT or STOP instruction.

4. If the internal oscillator is operating before the STOP mode is set, oscillation of the internal oscillation clock cannot be stopped in the STOP mode. However, when the internal oscillation clock is used as the CPU clock, the CPU operation is stopped for 17/f_R (s) after STOP mode is released.

20.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

This is the status register of the X1 input clock oscillation stabilization time counter. If the internal oscillation clock is used as the CPU clock, the X1 input clock oscillation stabilization time can be checked.

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset release (reset by $\overline{\text{RESET}}$ input, POC, LVI, and WDT), the STOP instruction, and MSTOP (bit 7 of MOC register) = 1 clear OSTC to 00H.

Figure 20-1. Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFA3H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
OSTC	0	0	0	MOST11	MOST13	MOST14	MOST15	MOST16

MOST11	MOST13	MOST14	MOST15	MOST16	Oscillation stabilization time status	
						f_{XP} = 20MHz
1	0	0	0	0	$2^{11}/f_{XP}$ min.	102.4 μs min.
1	1	0	0	0	$2^{13}/f_{XP}$ min.	409.6 μs min.
1	1	1	0	0	$2^{14}/f_{XP}$ min.	819.2 μs min.
1	1	1	1	0	$2^{15}/f_{XP}$ min.	1.64 ms min.
1	1	1	1	1	$2^{16}/f_{XP}$ min.	3.27 ms min.

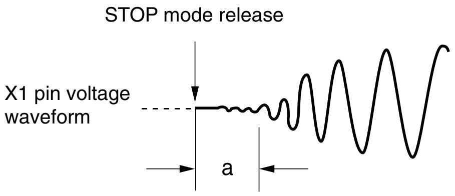
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST11 and remain 1.

2. If the STOP mode is entered and then released while the internal oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The X1 oscillation stabilization time counter counts only during the oscillation stabilization time set by OSTS. Therefore, note that only the statuses during the oscillation stabilization time set by OSTS are set to OSTC after STOP mode has been released.

3. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts ("a" below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_X: X1 input clock oscillation frequency

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 oscillation stabilization wait time when STOP mode is released. The wait time set by OSTS is valid only after STOP mode is released when the X1 input clock is selected as the CPU clock. After STOP mode is released when the internal oscillation clock is selected, check the oscillation stabilization time using OSTC.

OSTS can be set by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets OSTS to 05H.

Figure 20-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFA4H After reset: 05H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

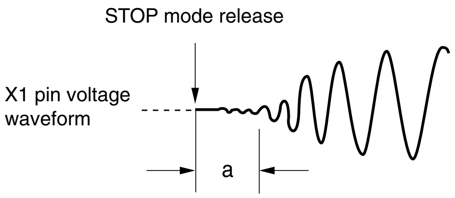
OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection	
				f_{XP} = 20MHz
0	0	1	$2^{11}/f_{XP}$	102.4 μs
0	1	0	$2^{13}/f_{XP}$	409.6 μs
0	1	1	$2^{14}/f_{XP}$	819.2 μs
1	0	0	$2^{15}/f_{XP}$	1.64 ms
1	0	1	$2^{16}/f_{XP}$	3.27 ms
Other than above			Setting prohibited	

Cautions 1. If the STOP mode is entered and then released while the internal oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

The X1 oscillation stabilization time counter counts only during the oscillation stabilization time set by OSTS. Therefore, note that only the statuses during the oscillation stabilization time set by OSTS are set to OSTC after STOP mode has been released.

2. The wait time when STOP mode is released does not include the time after STOP mode release until clock oscillation starts (“a” below) regardless of whether STOP mode is released by $\overline{\text{RESET}}$ input or interrupt generation.



Remark f_X: X1 input clock oscillation frequency

20.2 Standby Function Operation

20.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the X1 input clock or internal oscillation clock.

The operating statuses in the HALT mode are shown below.

Table 20-2. Operating Statuses in HALT Mode

HALT Mode Setting		When HALT Instruction Is Executed While CPU Is Operating on X1 Input Clock		When HALT Instruction Is Executed While CPU Is Operating on Internal Oscillation Clock	
Item		When Internal Oscillation Clock Continues	When Internal Oscillation Clock Stopped[Note 1]	When X1 Input Clock Oscillation Continues	When X1 Input Clock Oscillation Stopped
System clock		Clock supply to the CPU is stopped.			
CPU		Operation stopped			
Port (latch)		Status before HALT mode was set is retained			
10-bit inverter control timer		Operable		Operation not guaranteed	
16-bit up/down counter ITENC20		Operable		Operation not guaranteed	
16-bit timer/event counter 00		Operable		Operation not guaranteed	
8-bit timer/event counter 50		Operable		Operation not guaranteed when count clock other than TI50 is selected	
8-bit timer/event counter 51		Operable		Operation not guaranteed when count clock other than $f_R/2^7$ or TI51 is selected	
8-bit timer H0		Operable		Operation not guaranteed when count clock other than TM50 output is selected during 8-bit timer/event counter 50 operation	
Watchdog timer	Internal oscillator cannot be stopped[Note 2]	Operable	–	Operable	
	Internal oscillator can be stopped[Note 2]	Operation stopped			
Clock output/buzzer output controller		Operable			Operation not guranteed
Real-time output ports		Operable		Operation not guaranteed when other than external trigger (INTP2) is used or when other than TI51 is selected as count clock of 8-bit timer/event counter 51.	
A/D converter		Operable		Operation not guaranteed	
Serial interface	UART00	Operable		Operation not guaranteed when serial clock other than TM50 output is selected during 8-bit timer/event counter 50 operation	
	CSI10	Operable		Operation not guaranteed when serial clock other than external $\overline{\text{SCK10}}$ is selected	
Multiplier/divider		Operable		Operation not guaranteed	
Power-on-clear function		Operable			
Low-voltage detection function		Operable			
External interrupt		Operable			

Notes 1. When “Stopped by software” is selected for internal oscillator by an option byte and internal oscillator is stopped by software (for option bytes, see **CHAPTER 24 OPTION BYTES**).

2. “Internal oscillator cannot be stopped” or “Internal oscillator can be stopped by software” can be selected by an option byte.

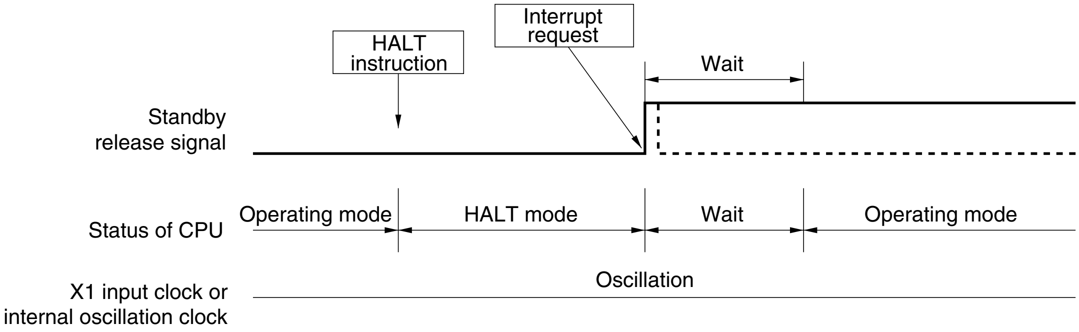
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgement is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgement is disabled, the next address instruction is executed.

Figure 20-3. HALT Mode Release by Interrupt Request Generation



Remarks 1. The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

2. The wait time is as follows:

- When vectored interrupt servicing is carried out: 8 or 9 clocks
- When vectored interrupt servicing is not carried out: 2 or 3 clocks

<R>

(b) Release by non-maskable interrupt request

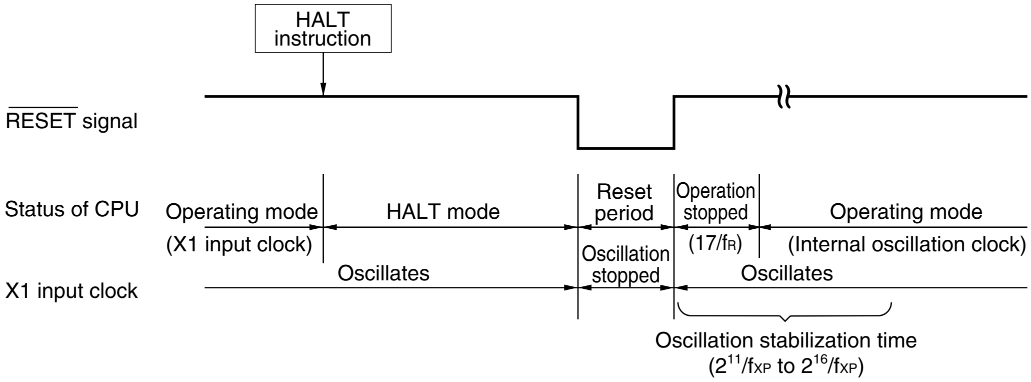
When a non-maskable interrupt request is generated, the HALT mode is released and vectored interrupt servicing is carried out whether interrupt acknowledgment is enabled or disabled.

(c) Release by $\overline{\text{RESET}}$ input

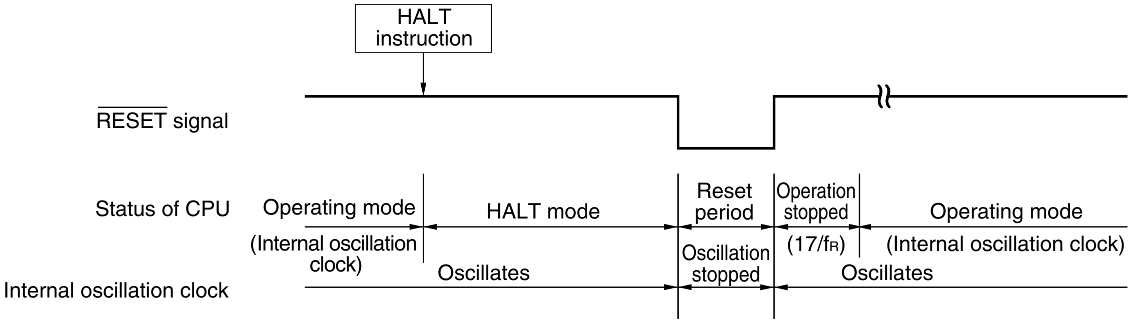
When the $\overline{\text{RESET}}$ signal is input, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 20-4. HALT Mode Release by $\overline{\text{RESET}}$ Input

(1) When X1 input clock is used as CPU clock


HALT instruction
$\overline{\text{RESET}}$ signal
Status of CPU
Operating mode (X1 input clock)
HALT mode
Reset period
Operation stopped (17/fR)
Operating mode (Internal oscillation clock)
X1 input clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (2^{11}/fXP to 2^{16}/fXP)

(2) When internal oscillation clock is used as CPU clock


HALT instruction
$\overline{\text{RESET}}$ signal
Status of CPU
Operating mode (Internal oscillation clock)
HALT mode
Reset period
Operation stopped (17/fR)
Operating mode (Internal oscillation clock)
Internal oscillation clock
Oscillates
Oscillation stopped
Oscillates

Remarks 1. f_{XP}: X1 input clock oscillation frequency

2. f_R: Internal oscillation clock frequency

Table 20-3. Operation in Response to Interrupt Request in HALT Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	HALT mode held
<R> Non-maskable interrupt request	–	–	×	×	Interrupt servicing execution
$\overline{\text{RESET}}$ input	–	–	×	×	Reset processing

×: don't care

20.2.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set regardless of whether the CPU clock before the setting was the X1 input clock or internal oscillation clock.

Caution Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

The operating statuses in the STOP mode are shown below.

Table 20-4. Operating Statuses in STOP Mode

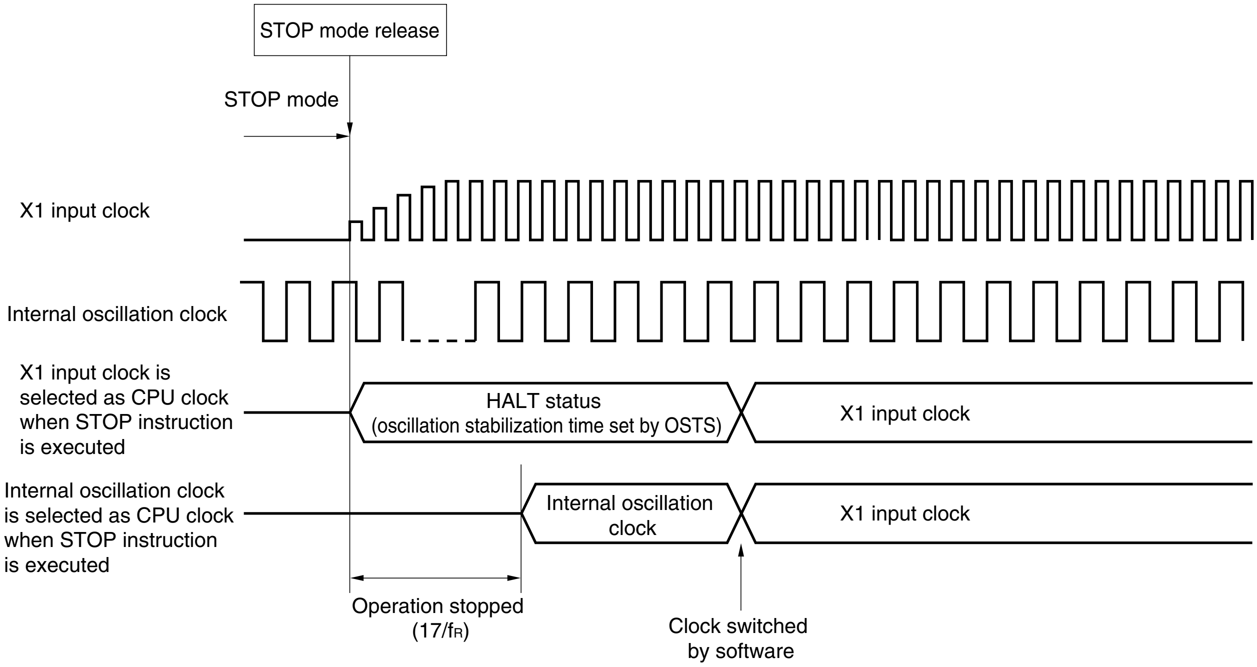
STOP Mode Setting Item		When STOP Instruction Is Executed While CPU Is Operating on X1 Input Clock		When STOP Instruction Is Executed While CPU Is Operating on Internal Oscillation Clock
		When Internal Oscillation Clock Continues	When Internal Oscillation Clock Stopped[Note 1]	
System clock		Only X1 oscillator oscillation is stopped. Clock supply to the CPU is stopped.		
CPU		Operation stopped		
Port (latch)		Status before STOP mode was set is retained		
10-bit inverter control timer		Operation stopped		
16-bit up/down counter ITENC20		Operation stopped		
16-bit timer/event counter 00		Operation stopped		
8-bit timer/event counter 50		Operable only when TI50 is selected as the count clock		
8-bit timer/event counter 51		Operable only when TI51 is selected as the count clock		
8-bit timer H0		Operable only when TM50 output is selected as the count clock during 8-bit timer/event counter 50 operation		
Watchdog timer	Internal oscillator cannot be stopped[Note 2]	Operable	–	Operable
	Internal oscillator can be stopped[Note 2]	Operation stopped		
Clock output/buzzer output controller		Operation stopped		
Real-time output ports		Operable only when INTTM51 is selected as the count clock during TM51 operation or when external trigger (INTP2) is selected		
A/D converter		Operation stopped		
Serial interface	UART00	Operable only when TM50 output is selected as the count clock during TM50 operation		
	CSI10	Operable only when external $\overline{\text{SCK10}}$ is selected as the serial clock		
Multiplier/divider		Operation stopped		
Power-on-clear function		Operable		
Low-voltage detection function		Operable		
External interrupt		Operable		

Notes 1. When “Stopped by software” is selected for internal oscillator by an option byte and internal oscillator is stopped by software (for option bytes, see **CHAPTER 24 OPTION BYTES**).

2. “Internal oscillator cannot be stopped” or “Internal oscillator can be stopped by software” can be selected by an option byte.

(2) STOP mode release

Figure 20-5. Operation Timing When STOP Mode Is Released


STOP mode release
STOP mode
X1 input clock
Internal oscillation clock
X1 input clock is selected as CPU clock when STOP instruction is executed
HALT status
(oscillation stabilization time set by OSTS)
X1 input clock
Internal oscillation clock is selected as CPU clock when STOP instruction is executed
Internal oscillation clock
X1 input clock
Operation stopped
(17/fR)
Clock switched by software

The STOP mode can be released by the following three sources.

(a) Release by unmasked interrupt request

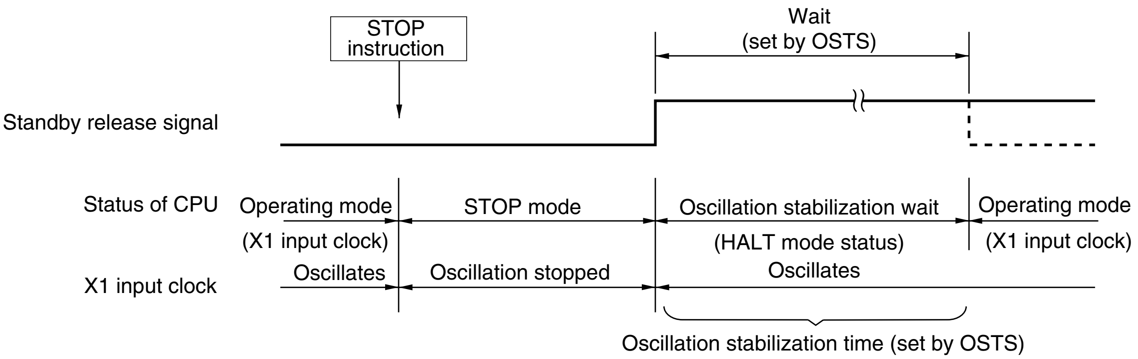
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

<R> **(b) Release by non-maskable interrupt request**

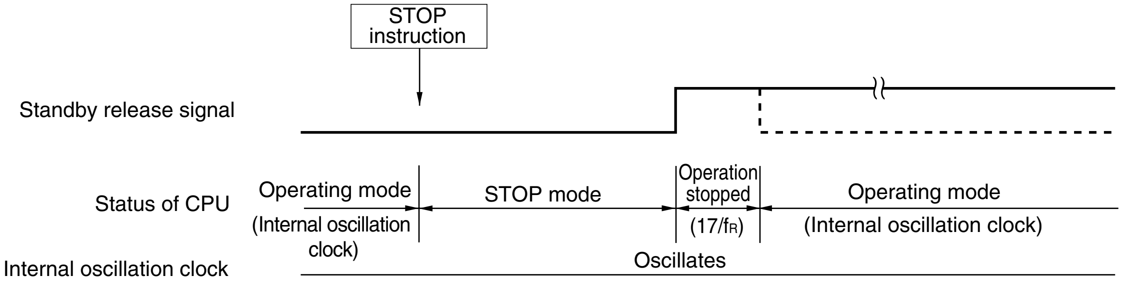
When a non-maskable interrupt request is generated, the STOP mode is released whether interrupt acknowledgment is enabled or disabled. And after the oscillation stabilization time has elapsed, vectored interrupt servicing is carried out.

Figure 20-6. STOP Mode Release by Interrupt Request Generation

(1) When X1 input clock is used as CPU clock


STOP instruction
Wait (set by OSTS)
Standby release signal
Status of CPU
Operating mode (X1 input clock)
STOP mode
Oscillation stabilization wait (HALT mode status)
Operating mode (X1 input clock)
X1 input clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stabilization time (set by OSTS)

(2) When internal oscillation clock is used as CPU clock


STOP instruction
Standby release signal
Status of CPU
Operating mode (Internal oscillation clock)
STOP mode
Operation stopped (17/fR)
Operating mode (Internal oscillation clock)
Internal oscillation clock
Oscillates

Remarks 1. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

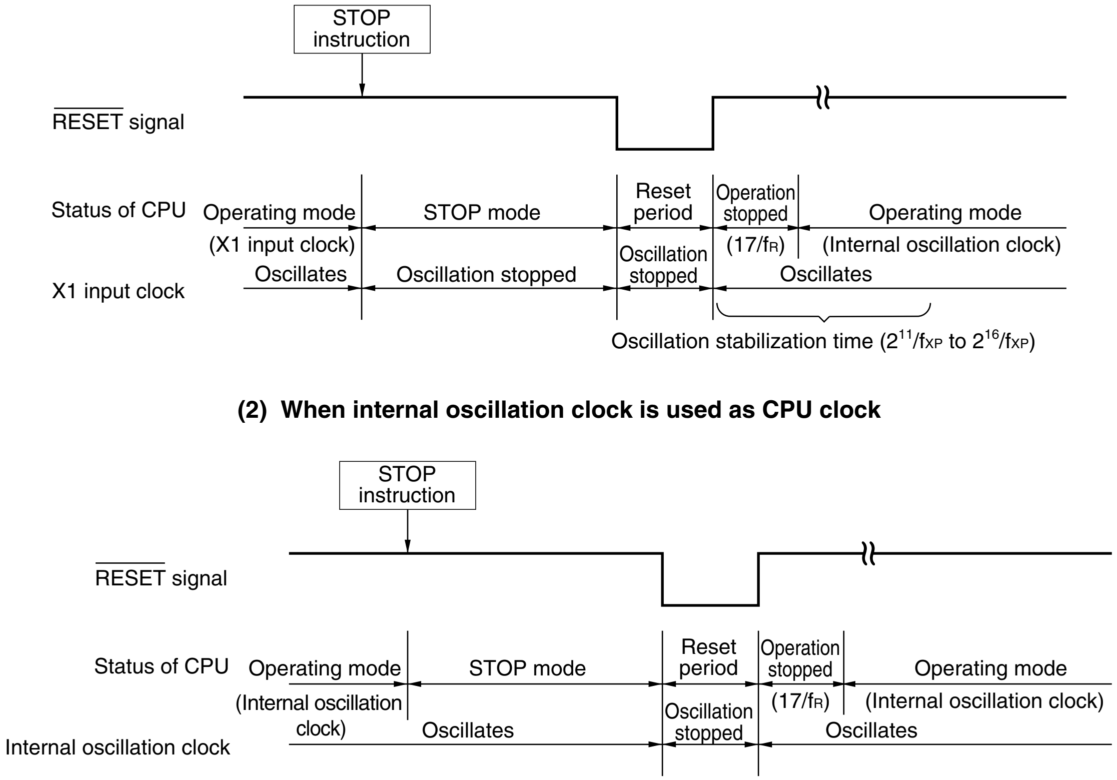
2. f_R: Internal oscillation clock frequency

(c) Release by $\overline{\text{RESET}}$ input

When the $\overline{\text{RESET}}$ signal is input, STOP mode is released and a reset operation is performed after the oscillation stabilization time has elapsed.

Figure 20-7. STOP Mode Release by $\overline{\text{RESET}}$ Input

(1) When X1 input clock is used as CPU clock


STOP instruction
RESET signal
Status of CPU
Operating mode (X1 input clock)
STOP mode
Reset period
Operation stopped (17/fR)
Operating mode (Internal oscillation clock)
X1 input clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (2^11/fXP to 2^16/fXP)
STOP instruction
RESET signal
Status of CPU
Operating mode (Internal oscillation clock)
STOP mode
Reset period
Operation stopped (17/fR)
Operating mode (Internal oscillation clock)
Internal oscillation clock
Oscillates
Oscillation stopped
Oscillates

(2) When internal oscillation clock is used as CPU clock

Remarks 1. f_{XP}: X1 input clock oscillation frequency

2. f_R: Internal oscillation clock frequency

Table 20-5. Operation in Response to Interrupt Request in STOP Mode

Release Source	MK××	PR××	IE	ISP	Operation
Maskable interrupt request	0	0	0	×	Next address instruction execution
	0	0	1	×	Interrupt servicing execution
	0	1	0	1	Next address instruction execution
	0	1	×	0	
	0	1	1	1	Interrupt servicing execution
	1	×	×	×	STOP mode held
<R> Non-maskable interrupt request	−	−	×	×	Interrupt servicing execution
$\overline{\text{RESET}}$ input	−	−	×	×	Reset processing

×: don’t care

CHAPTER 21 RESET FUNCTION

The following four operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-clear (POC) circuit
(4) Internal reset by comparison of supply voltage and detection voltage of low-power-supply detector (LVI)

External and internal resets have no functional differences. In both cases, program execution starts at the address at 0000H and 0001H when the reset signal is input.

A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POC and LVI circuit voltage detection, and each item of hardware is set to the status shown in Table 21-1. Each port pin is high impedance during reset input or during the oscillation stabilization time just after reset release.

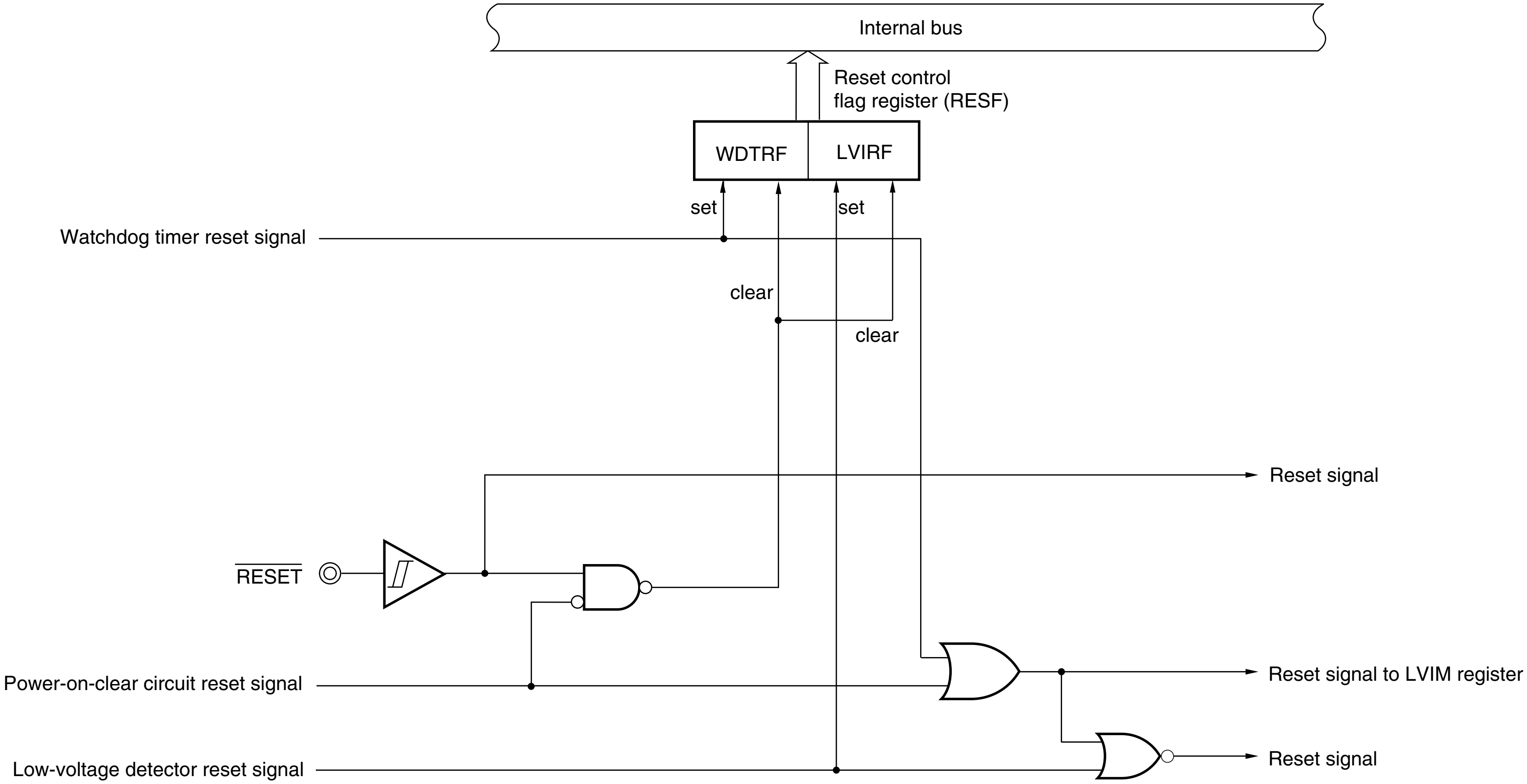
When a high level is input to the $\overline{\text{RESET}}$ pin, the reset is released and program execution starts using the internal oscillation clock after the CPU clock operation has stopped for $17/f_R$ (s). A reset generated by the watchdog timer is automatically released after the reset, and program execution starts using the internal oscillation clock after the CPU clock operation has stopped for $17/f_R$ (s) (see **Figures 21-2** to **21-4**). Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} > V_{POC}$ or $V_{DD} > V_{LVI}$ after the reset, and program execution starts using the internal oscillation clock after the CPU clock operation has stopped for $17/f_R$ (s) (see **CHAPTER 22 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 23 LOW-VOLTAGE DETECTOR**).

Cautions 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.

2. During reset input, the X1 input clock and internal oscillation clock stop oscillating.

3. When the STOP mode is released by a reset, the STOP mode contents are held during reset input. However, the port pins become high-impedance.

Figure 21-1. Block Diagram of Reset Function


Internal bus
Reset control
flag register (RESF)
WDTRF
LVIRF
set
set
Watchdog timer reset signal
clear
clear
Reset signal
RESET
Power-on-clear circuit reset signal
Reset signal to LVIM register
Low-voltage detector reset signal
Reset signal

Caution **An LVI circuit internal reset does not reset the LVI circuit.**

Remark LVIM: Low-voltage detection register

Figure 21-2. Timing of Reset by $\overline{\text{RESET}}$ Input

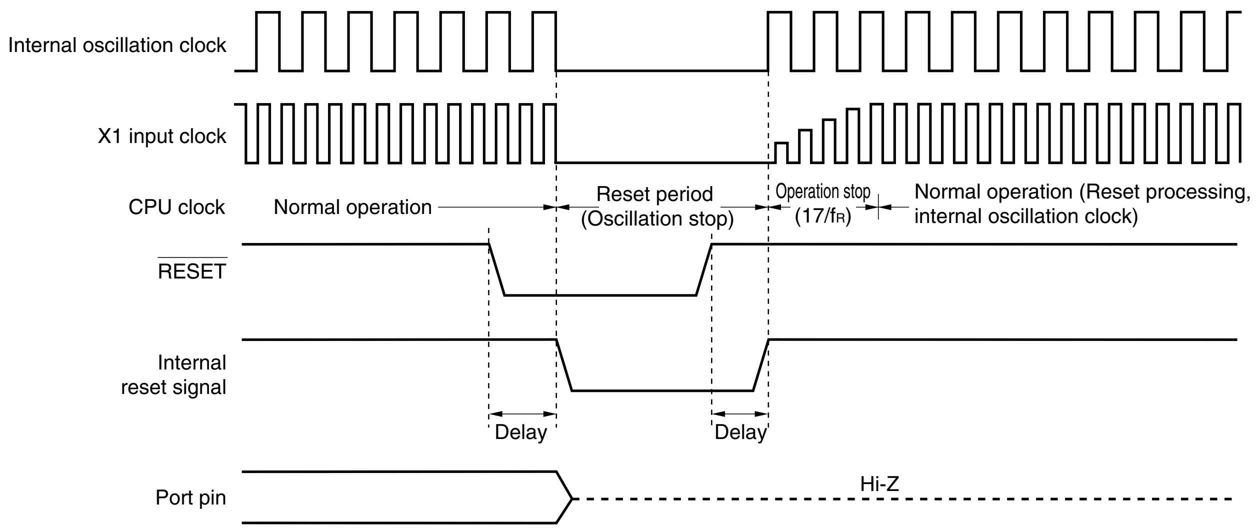
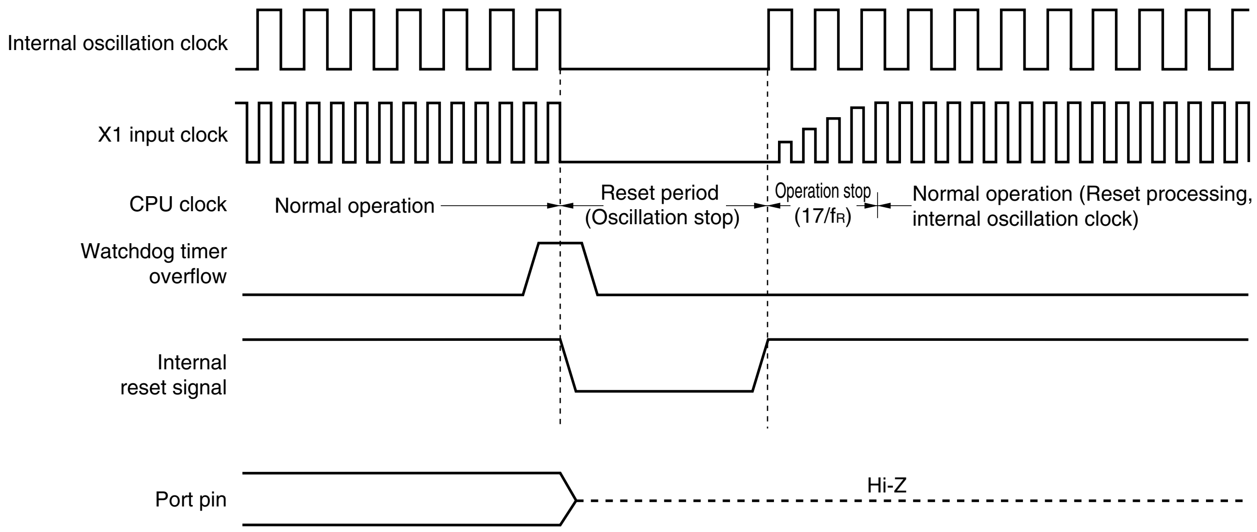
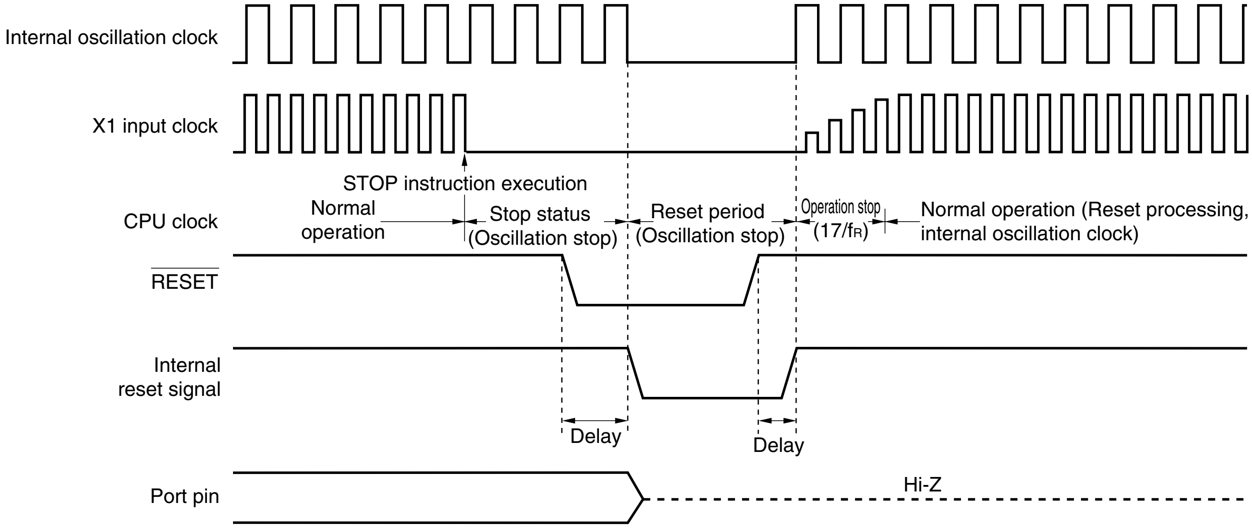

Internal oscillation clock
X1 input clock
CPU clock
Normal operation
Reset period (Oscillation stop)
Operation stop (17/fR)
Normal operation (Reset processing, internal oscillation clock)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Figure 21-3. Timing of Reset Due to Watchdog Timer Overflow


Internal oscillation clock
X1 input clock
CPU clock
Normal operation
Reset period (Oscillation stop)
Operation stop (17/fR)
Normal operation (Reset processing, internal oscillation clock)
Watchdog timer overflow
Internal reset signal
Port pin
Hi-Z

Caution A watchdog timer internal reset resets the watchdog timer.

Figure 21-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input


Internal oscillation clock
X1 input clock
STOP instruction execution
CPU clock
Normal operation
Stop status (Oscillation stop)
Reset period (Oscillation stop)
Operation stop (17/fR)
Normal operation (Reset processing, internal oscillation clock)
RESET
Internal reset signal
Delay
Delay
Port pin
Hi-Z

Remark For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 22 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 23 LOW-VOLTAGE DETECTOR**.

Table 21-1. Hardware Statuses after Reset Acknowledgment (1/3)

Hardware		Status After Reset Acknowledgment[Note 1]
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		02H
RAM	Data memory	Undefined[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P7) (output latches)		00H (undefined only for P2)
Port mode registers (PM0, PM1, PM3 to PM7)		FFH
Pull-up resistor option registers (PU0, PU1, PU3 to PU7)		00H
Internal memory size switching register (IMS)		CFH
Processor clock control register (PCC)		00H
Internal oscillation mode register (RCM)		00H
Main clock mode register (MCM)		00H
Main OSC control register (MOC)		00H
Oscillation stabilization time select register (OSTS)		05H
Oscillation stabilization time counter status register (OSTC)		00H
System wait control register (VSWC)		00H
10-bit inverter control timer	Compare registers (TW0CM0 to TW0CM2, TW0CM4, TW0CM5)	000H
	Compare register (TW0CM3)	0FFH
	Buffer registers (TW0BFCM0 to TW0BFCM2, TW0BFCM4, TW0BFCM5)	000H
	Buffer register (TW0BFCM3)	0FFH
	Dead time reload register (TW0DTIME)	FFH
	Control register (TW0C)	00H
	Mode register (TW0M)	00H
	A/D trigger select register (TW0TRGS)	00H
	Output control register (TW0OC)	00H
16-bit up/down counter ITENC20	Up/down counter (IT20UDC)	0000H
	Compare registers 0, 1 (IT20CM0, IT20CM1)	0000H
	Capture/compare registers 0, 1 (IT20CC0, IT20CC1)	0000H
	Unit mode register (IT20TUM)	00H
	Control register (IT20TMC)	00H
	Capture/compare control register (IT20CCR)	00H
	Effective edge select register (IT20SESA)	00H
	Prescaler mode register (IT20PRM)	07H
	Status register (IT20STS)	00H

Notes 1. During reset input or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Table 21-1. Hardware Statuses after Reset Acknowledgment (2/3)

Hardware		Status After Reset Acknowledgment
16-bit timer/event counter 00	Timer counter 00 (TM00)	0000H
	Capture/compare registers 00, 01 (CR00, CR01)	0000H
	Mode control register 00 (TMC00)	00H
	Prescaler mode register 00 (PRM00)	00H
	Capture/compare control register 00 (CRC00)	00H
	Timer output control register 00 (TOC00)	00H
8-bit timer/event counters 50, 51	Timer counters 50, 51 (TM50, TM51)	00H
	Compare registers 50, 51 (CR50, CR51)	00H
	Timer clock selection registers 50, 51 (TCL50, TCL51)	00H
	Mode control registers 50, 51 (TMC50, TMC51)	00H
8-bit timer H0	Compare registers 00, 01 (CMP00, CMP01)	00H
	Mode register (TMHMD0)	00H
Clock output/buzzer output controller	Clock output selection register (CKS)	00H
Watchdog timer	Mode register (WDTM)	67H
	Enable register (WDTE)	9AH
Real-time output ports	Buffer registers (RTBL00, RTBH00, RTBL01, RTBH01)	00H
	Mode registers (RTPM00, RTPM01)	00H
	Control registers (RTPC00, RTPC01)	00H
	DC control registers (DCCTL00, DCCTL01)	00H
A/D converter	Conversion result register (ADCR)	Undefined
	Mode register (ADM)	00H
	Analog input channel specification register (ADS)	00H
	Power-fail comparison mode register (PFM)	00H
	Power-fail comparison threshold register (PFT)	00H
Serial interface UART00	Receive buffer register 00 (RXB00)	FFH
	Transmit shift register 00 (TXS00)	FFH
	Asynchronous serial interface operation mode register 00 (ASIM00)	01H
	Baud rate generator control register 00 (BRGC00)	1FH
Serial interface CSI10	Transmit buffer register 10 (SOTB10)	Undefined
	Serial I/O shift register 10 (SIO10)	00H
	Serial operation mode register 10 (CSIM10)	00H
	Serial clock selection register 10 (CSIC10)	00H
Multiplier/divider	Remainder data register 0 (SDR0)	0000H
	Multiplication/division data register A0 (MDA0H, MDA0L)	0000H
	Multiplication/division data register B0 (MDB0)	0000H
	Multiplier/divider control register 0 (DMUC0)	00H

Table 21-1. Hardware Statuses After Reset Acknowledgment (3/3)

Hardware		Status After Reset Acknowledgment
Reset function	Reset control flag register (RESF)	00H[Note 1]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 1]
Interrupt	Request flag registers 0L, 0H, 1L, 1H (IF0L, IF0H, IF1L, IF1H)	00H
	Mask flag registers 0L, 0H, 1L (MK0L, MK0H, MK1L)	FFH
	Mask flag register 1H (MK1H)	DFH
	Priority specification flag registers 0L, 0H, 1L, 1H (PR0L, PR0H, PR1L, PR1H)	FFH
	External interrupt rising edge enable register (EGP)	00H
	External interrupt falling edge enable register (EGN)	00H
Flash memory	Flash protect command register (PFCMD)	Undefined
	Flash status register (PFS)	00H
	Flash programming mode control register (FLPMC)	0XH[Note 2]

Notes 1. These values vary depending on the reset source.

Reset Source Register	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
RESF	See **Table 21-2.**			
LVIM	Cleared (00H)	Cleared (00H)	Cleared (00H)	Held

2. This value varies depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

21.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the μPD78F0714. The reset control flag register (RESF) is used to store which source has generated the reset request.

RESF can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset input by power-on-clear (POC) circuit, and reading RESF clear RESF to 00H.

Figure 21-5. Format of Reset Control Flag Register (RESF)

Address: FFACH After reset: 00H[Note] R

Symbol	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LVIRF

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Note The value after reset varies depending on the reset source.

Caution Do not read data by a 1-bit memory manipulation instruction.

The status of RESF when a reset request is generated is shown in Table 21-2.

Table 21-2. RESF Status When Reset Request Is Generated

Reset Source Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by WDT	Reset by LVI
WDTRF	Cleared (0)	Cleared (0)	Set (1)	Held
LVIRF			Held	Set (1)

CHAPTER 22 POWER-ON-CLEAR CIRCUIT

22.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit (POC) has the following functions.

- Generates internal reset signal at power on.
- Compares supply voltage (V_{DD}) and detection voltage (V_{POC} = 3.5 V ±0.2 V[Note]), and generates internal reset signal when $V_{DD} < V_{POC}$.

Note This value may change after evaluation.

Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

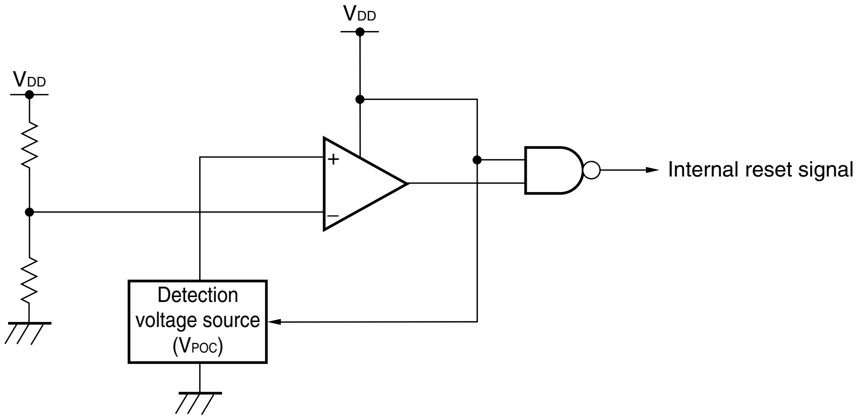
Remark This product incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset cause is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), or low-voltage-detection (LVI) circuit. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT, or LVI.

For details of the RESF, see **CHAPTER 21 RESET FUNCTION**.

22.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 22-1.

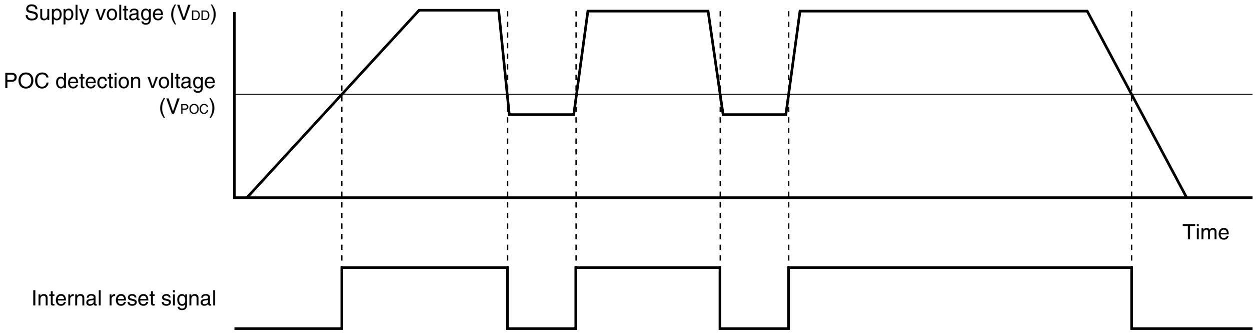
Figure 22-1. Block Diagram of Power-on-Clear Circuit



22.3 Operation of Power-on-Clear Circuit

In the power-on-clear circuit, the supply voltage (V_{DD}) and detection voltage (V_{POC}) are compared, and when V_{DD} < V_{POC}, an internal reset signal is generated.

Figure 22-2. Timing of Internal Reset Signal Generation in Power-on-Clear Circuit



22.4 Cautions for Power-on-Clear Circuit

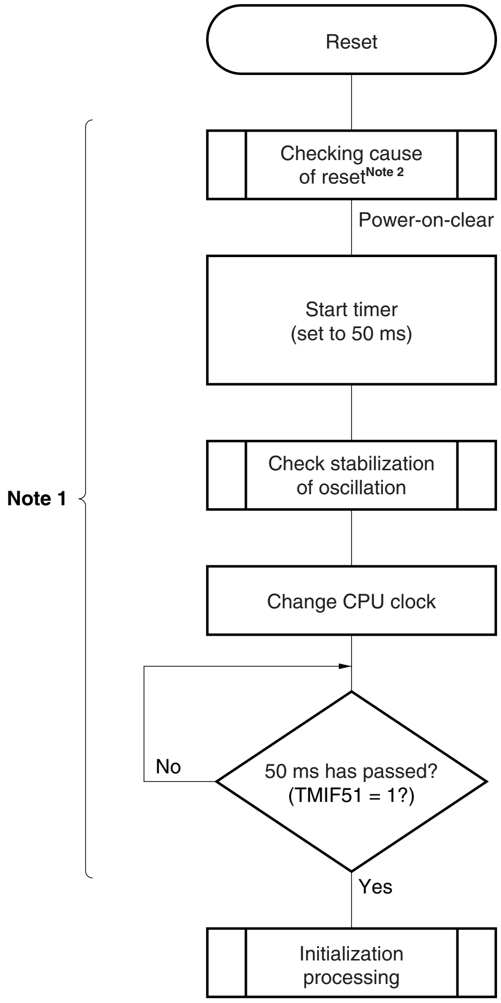
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POC}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 22-3. Example of Software Processing After Release of Reset (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage


Reset
Checking cause of resetNote 2
Power-on-clear
Start timer (set to 50 ms)
Check stabilization of oscillation
Change CPU clock
No
50 ms has passed? (TMIF51 = 1?)
Yes
Initialization processing
Note 1

; The internal oscillation clock is set as the CPU clock when the reset signal is generated

; The cause of reset (power-on-clear, WDT, or LVI) can be identified by the RESF register.

; 8-bit timer 51 can operate with the internal oscillation clock.
Source: f_R (480 kHz (MAX.))/2^7 × compare value 200 = 53 ms
(f_R: Internal oscillation clock frequency)

; Check the stabilization of oscillation of the X1 input clock by using the OSTC register.

; Change the CPU clock from the internal oscillation clock to the X1 input clock.

; TMIF51 = 1: Interrupt request is generated.

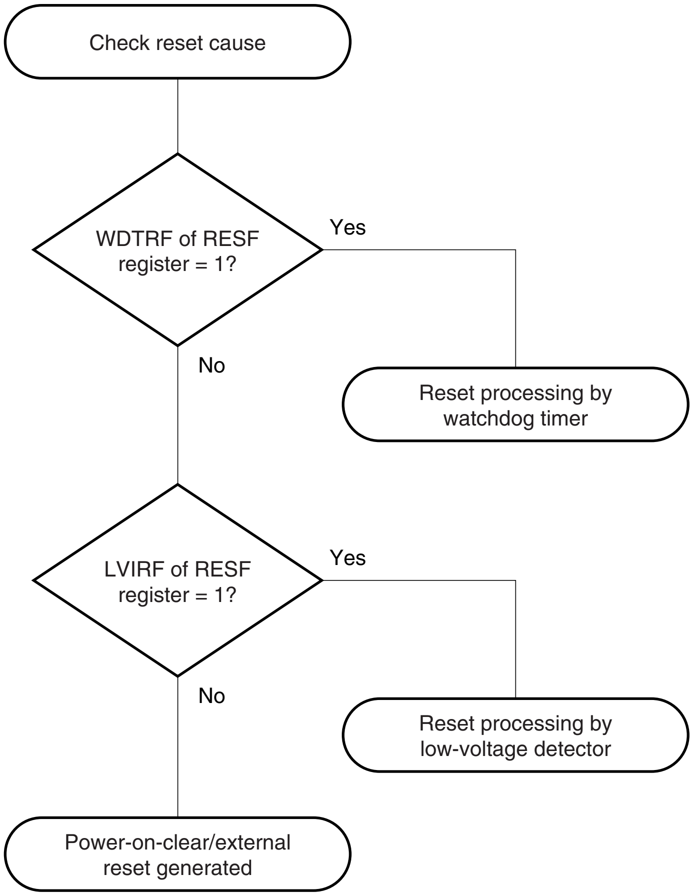
; Initialization of ports

Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 22-3. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause


Check reset cause
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
LVIRF of RESF register = 1?
Yes
No
Reset processing by low-voltage detector
Power-on-clear/external reset generated

CHAPTER 23 LOW-VOLTAGE DETECTOR

23.1 Functions of Low-Voltage Detector

The low-voltage detector (LVI) has following functions.

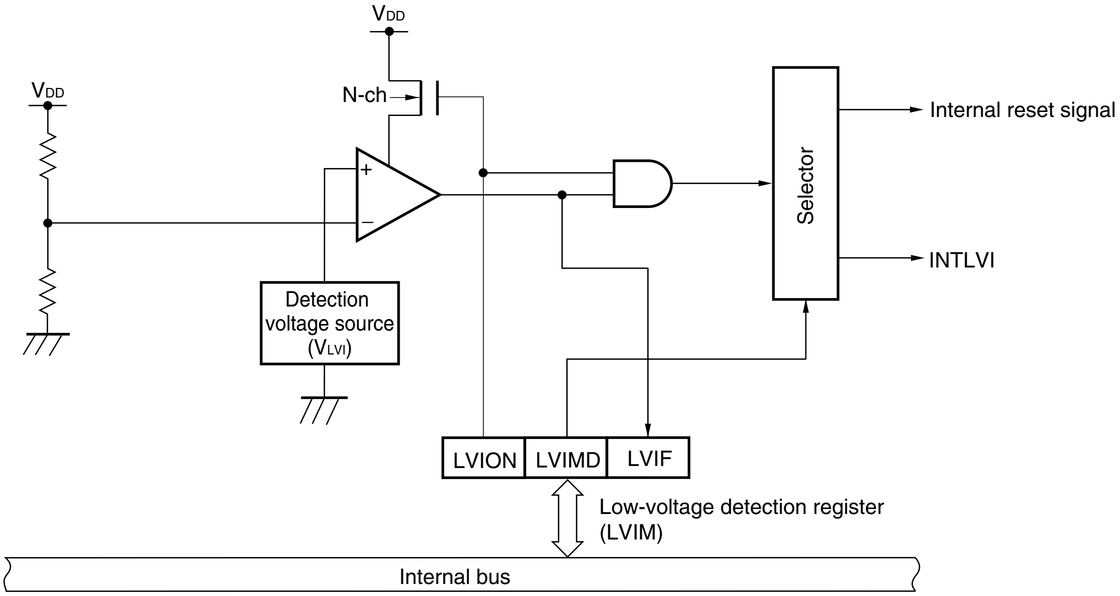
- Compares supply voltage (V_{DD}) and detection voltage (V_{LVI} = 4.3 V ±0.2 V), and generates an non-maskable interrupt signal or internal reset signal when $V_{DD} < V_{LVI}$.
- Interrupt or reset function can be selected by software.
- Operable in STOP mode.

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

23.2 Configuration of Low-Voltage Detector

A block diagram of the low-voltage detector is shown below.

Figure 23-1. Block Diagram of Low-Voltage Detector



23.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following register.

- Low-voltage detection register (LVIM)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 23-2. Format of Low-Voltage Detection Register (LVIM)

Address: FF78H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION[Notes 2, 3]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVIMD[Note 2]	Low-voltage detection operation mode selection
0	Generates interrupt signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})
1	Generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})

LVIF[Note 4]	Low-voltage detection flag
0	Supply voltage (V_{DD}) > detection voltage (V_{LVI}), or when operation is disabled
1	Supply voltage (V_{DD}) < detection voltage (V_{LVI})

Notes 1. Bit 0 is read-only.

2. LVION and LVIMD are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

3. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to instigate a wait of at least 0.2 ms from when LVION is set to 1 until the voltage is confirmed at LVIF.

4. The value of LVIF is output as the interrupt request signal INTLVI when LVION = 1 and LVIMD = 0.

Caution To stop LVI, follow either of the procedures below.

- **When using 8-bit memory manipulation instruction: Write 00H to LVIM.**
- **When using 1-bit memory manipulation instruction: Clear LVION to 0.**

23.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

- Used as reset
  Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates an internal reset signal when $V_{DD} < V_{LVI}$.
- Used as interrupt
  Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), and generates a non-maskable interrupt signal (INTLVI) when $V_{DD} < V_{LVI}$.

The operation is set as follows.

<R> **(1) When used as reset**

- When starting operation

<1> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<2> Use software to instigate a wait of at least 0.2 ms.

<3> Wait until it is checked that (supply voltage (V_{DD}) > detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.

<4> Set bit 1 (LVIMD) of LVIM to 1 (generates internal reset signal when supply voltage (V_{DD}) < detection voltage (V_{LVI})).

Figure 23-3 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <4> above.

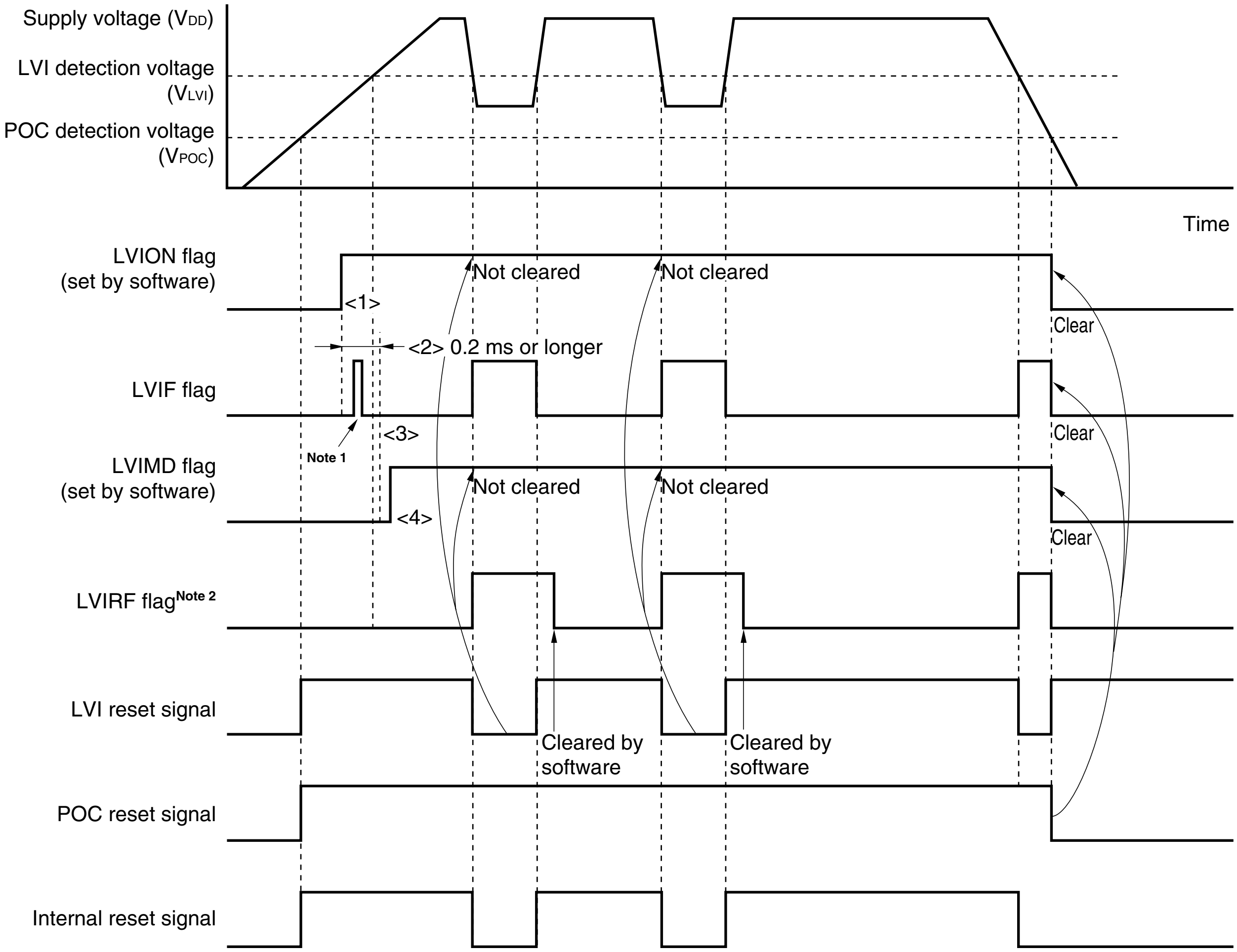
Cautions 1. A non-maskable interrupt signal (INTLVI) is generated, even if the low-voltage detector is used as a reset. Steps <2> to <4> must be added in the LVI interrupt servicing routine.

2. If supply voltage (V_{DD}) > detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation
  Either of the following procedures must be executed.
  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:
    Clear LVIMD to 0 first, and then clear LVION to 0.

<R>

Figure 23-3. Timing of Low-Voltage Detector Internal Reset Signal Generation



Notes 1. The LVIF flag may be set (1).

2. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 21 RESET FUNCTION**.

Remark <1> to <4> in Figure 23-3 above correspond to <1> to <4> in the description of “when starting operation” in **23.4 (1) When used as reset**.

<R> **(2) When used as interrupt**

- Before starting operation

  Define the flag (any name) that has the following meaning as the global variable, in advance.

  0: Checks that “supply voltage (V_{DD}) > detection voltage (V_{LVI})”, after LVI operation.
  1: Does not check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})”, after LVI operation.

- When starting operation

<1> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
<2> Flag judgment (0: Perform steps <4> to <6> shown below, 1: Perform step <3> and the subsequent steps shown below.)
<3> Use software to instigate a wait of at least 0.2 ms.
<4> Wait until it is checked that (supply voltage (V_{DD}) > detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.
<5> Clear the flag that was set before starting operation.
<6> Execute the EI instruction (when vectored interrupts are used).

- After low-voltage detection interrupt signal (INTLVI) generation

<1> Flag judgment (0: Perform steps <4> and <5> shown below as normal processing, 1: Perform step <2> and the subsequent steps shown below.)
<2> Use software to instigate a wait of at least 0.2 ms.
<3> Wait until it is checked that (supply voltage (V_{DD}) > detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.
<4> Execute the program to be performed when a low-voltage detection interrupt is generated.
<5> Return to the main routine.

Figure 23-4 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1>, <3>, and <4> of “when starting operation”.

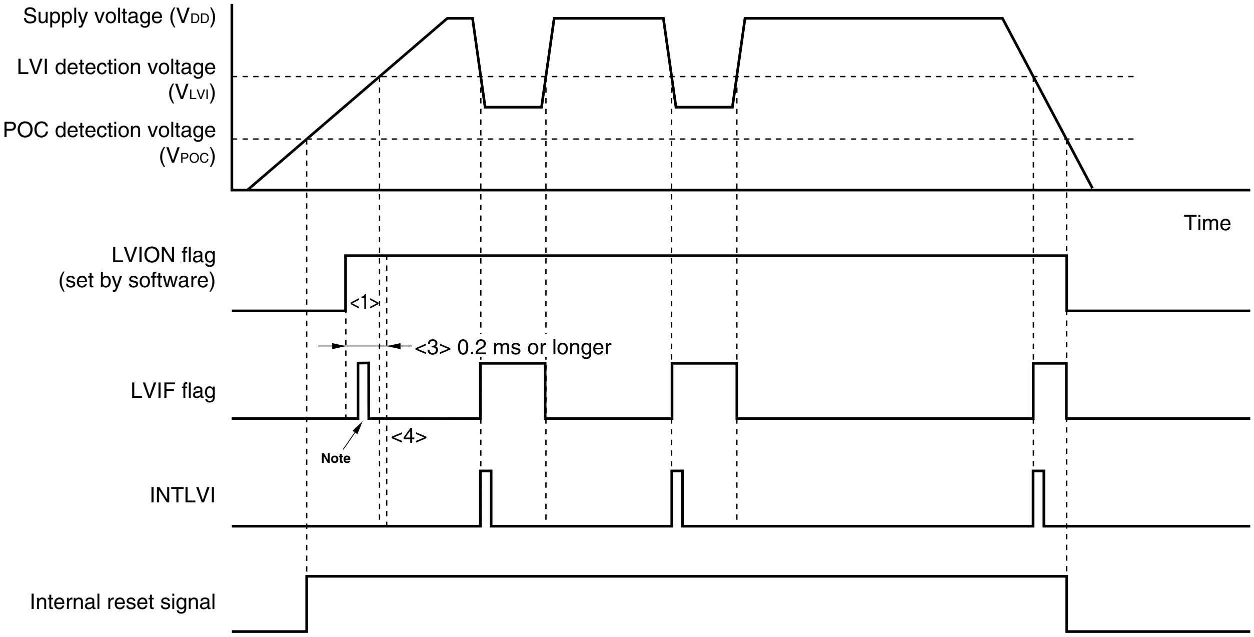
- When stopping operation

  Either of the following procedures must be executed.

  - When using 8-bit memory manipulation instruction:
    Write 00H to LVIM.
  - When using 1-bit memory manipulation instruction:
    Clear LVION to 0.

<R>

Figure 23-4. Timing of Low-Voltage Detector Interrupt Signal Generation


Supply voltage (V_{DD})
LVI detection voltage (V_{LVI})
POC detection voltage (V_{POC})
Time
LVION flag (set by software)
<1>
<3> 0.2 ms or longer
LVIF flag
Note
<4>
INTLVI
Internal reset signal

Note The LVIF flag may be set (1).

Remark <1>, <3>, and <4> in Figure 23-4 above correspond to <1>, <3>, and <4> in the description of “when starting operation” in **23.4 (2) When used as interrupt**.

23.5 Cautions for Low-Voltage Detector

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

(1) When used as reset

The system may be repeatedly reset and released from the reset status.

In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking action (1) below.

(2) When used as interrupt

Interrupt requests may be frequently generated. Take action (2) below.

In this system, take the following actions.

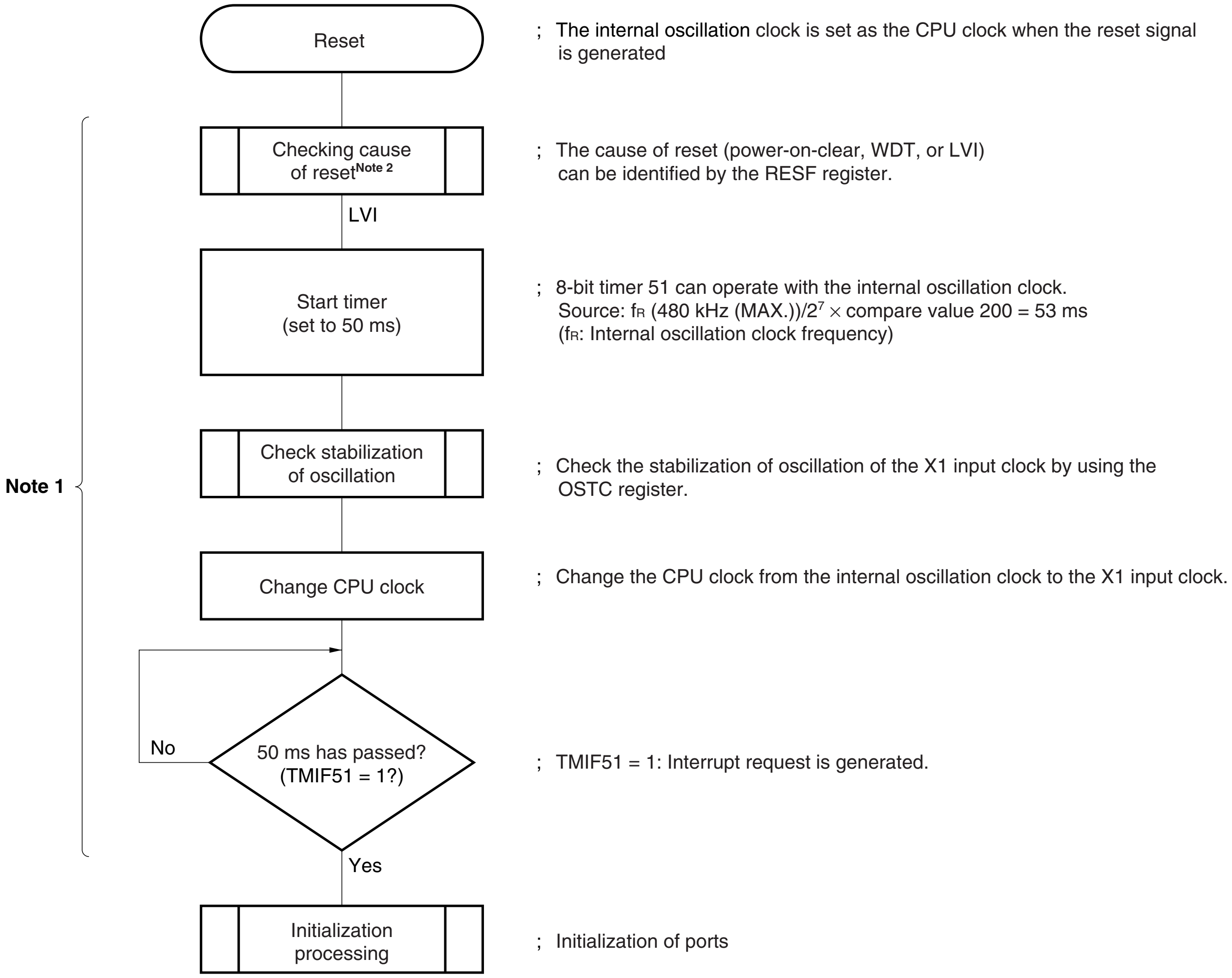
<Action>

(1) When used as reset

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 23-5. Example of Software Processing After Release of Reset (1/2)

- If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage


Reset
Checking cause of reset[Note 2]
LVI
Start timer (set to 50 ms)
Check stabilization of oscillation
Change CPU clock
50 ms has passed? (TMIF51 = 1?)
No
Yes
Initialization processing
Note 1

; The internal oscillation clock is set as the CPU clock when the reset signal is generated

; The cause of reset (power-on-clear, WDT, or LVI) can be identified by the RESF register.

; 8-bit timer 51 can operate with the internal oscillation clock.
Source: f_R (480 kHz (MAX.))/2^7 × compare value 200 = 53 ms
(f_R: Internal oscillation clock frequency)

; Check the stabilization of oscillation of the X1 input clock by using the OSTC register.

; Change the CPU clock from the internal oscillation clock to the X1 input clock.

; TMIF51 = 1: Interrupt request is generated.

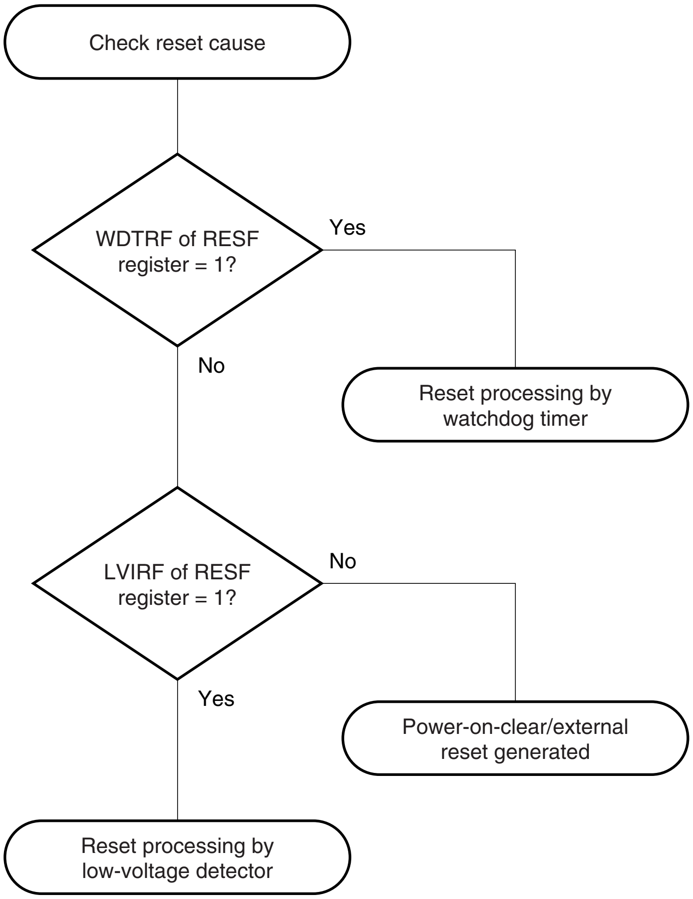
; Initialization of ports

Notes 1. If reset is generated again during this period, initialization processing is not started.

2. A flowchart is shown on the next page.

Figure 23-5. Example of Software Processing After Release of Reset (2/2)

- Checking reset cause


Check reset cause
WDTRF of RESF register = 1?
Yes
No
Reset processing by watchdog timer
LVIRF of RESF register = 1?
No
Yes
Power-on-clear/external reset generated
Reset processing by low-voltage detector

(2) When used as interrupt

Check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM).

In a system where the supply voltage fluctuation period is long in the vicinity of the LVI detection voltage, wait for the supply voltage fluctuation period, check that “supply voltage (V_{DD}) > detection voltage (V_{LVI})” using the LVIF flag.

CHAPTER 24 OPTION BYTES

The μPD78F0714 can realize selection to stop or enable internal oscillator with an option byte.

Option bytes are prepared at address 0080H in the flash memory.

When using flash memory version products, be sure to set to enable/disable to stop internal oscillator to the option bytes.

Figure 24-1. Allocation of Option Bytes

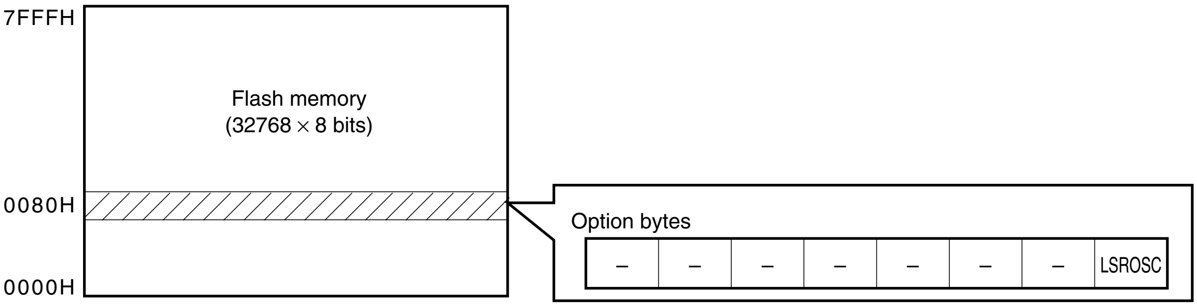


Figure 24-2. Format of Option Bytes

Address: 0080H

<R>

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	LSROSC

LSROSC	Internal oscillator operation
0	Can be stopped by software
1	Cannot be stopped

Cautions 1. To use the boot slip function, be sure to store the option data in the boot cluster 1 (for the boot swap function, see 25.8 Boot Swap Function).

2. Be sure to clear bits 1 to 7 to 0.

Remark An example of software coding for setting the option bytes is shown below.

```
OPT       CSEG     AT 0080H
OPTION:   DB       01H        ; Set to option byte (internal oscillator cannot be stopped)
```

CHAPTER 25 FLASH MEMORY

The μPD78F0714 with flash memory to which a program can be written, erased, and overwritten while mounted on the board.

25.1 Internal Memory Size Switching Register

The internal memory capacity set by using the internal memory size.
IMS is set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input sets IMS to CFH.

Caution Because the initial value of the memory size switching register (IMS) is CFH, set IMS to C8H by initialization.

Figure 25-1. Format of Internal Memory Size Switching Register (IMS)

Address: FFF0H After reset: CFH R/W

Symbol	7	6	5	4	3	2	1	0
IMS	RAM2	RAM1	RAM0	0	ROM3	ROM2	ROM1	ROM0

RAM2	RAM1	RAM0	Internal high-speed RAM capacity selection
1	1	0	1024 bytes
Other than above			Setting prohibited

ROM3	ROM2	ROM1	ROM0	Internal ROM capacity selection
1	0	0	0	32 KB
Other than above				Setting prohibited

25.2 Writing with Flash Memory Programmer

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the μPD78F0714 has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the μPD78F0714 is mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 25-1. Wiring Between μPD78F0714 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer			With CSI10		With CSI10 + HS		With UART00	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SO10/P17	40	SO10/P17	40	TxD00/P14	37
SO/TxD	Output	Transmit signal	SI10/P16	39	SI10/P16	39	RxD00/P13	36
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$/P15	38	$\overline{\text{SCK10}}$/P15	38	Not needed	Not needed
CLK	Output	Clock to μPD78F0714	X1	6	X1	6	X1	6
			X2[Note]	7	X2[Note]	7	X2[Note]	7
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	8	$\overline{\text{RESET}}$	8	$\overline{\text{RESET}}$	8
FLMD0	Output	Mode signal	FLMD0	3	FLMD0	3	FLMD0	3
FLMD1	Output	Mode signal	FLMD1/SO10/P17	40	FLMD1/SO10/P17	40	FLMD1/SO10/P17	40
H/S	Input	Handshake signal	–	–	P64	49	–	Not needed
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}	4	V_{DD}	4	V_{DD}	4
			EV_{DD}	26	EV_{DD}	26	EV_{DD}	26
			AV_{REF}	1	AV_{REF}	1	AV_{REF}	1
GND	–	Ground	V_{SS}	5	V_{SS}	5	V_{SS}	5
			EV_{SS}	25	EV_{SS}	25	EV_{SS}	25
			AV_{SS}	2	AV_{SS}	2	AV_{SS}	2

Note When using the clock out of the flash memory programmer, connect CLK of the programmer to X1, and connect its inverse signal to X2.

Examples of the recommended connection when using the adapter for flash memory writing are shown below.

Figure 25-2. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10) Mode

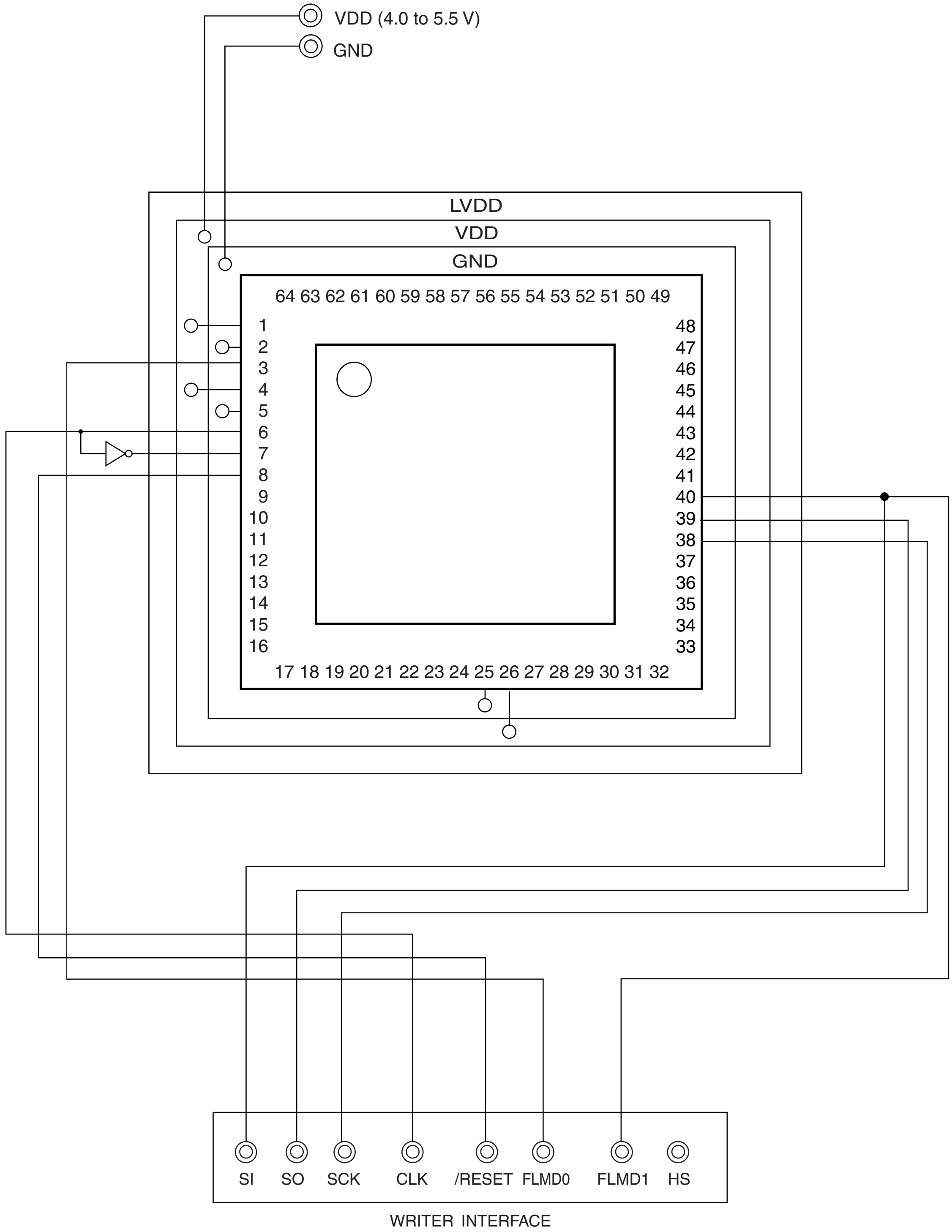


Figure 25-3. Example of Wiring Adapter for Flash Memory Writing in 3-Wire Serial I/O (CSI10 + HS) Mode

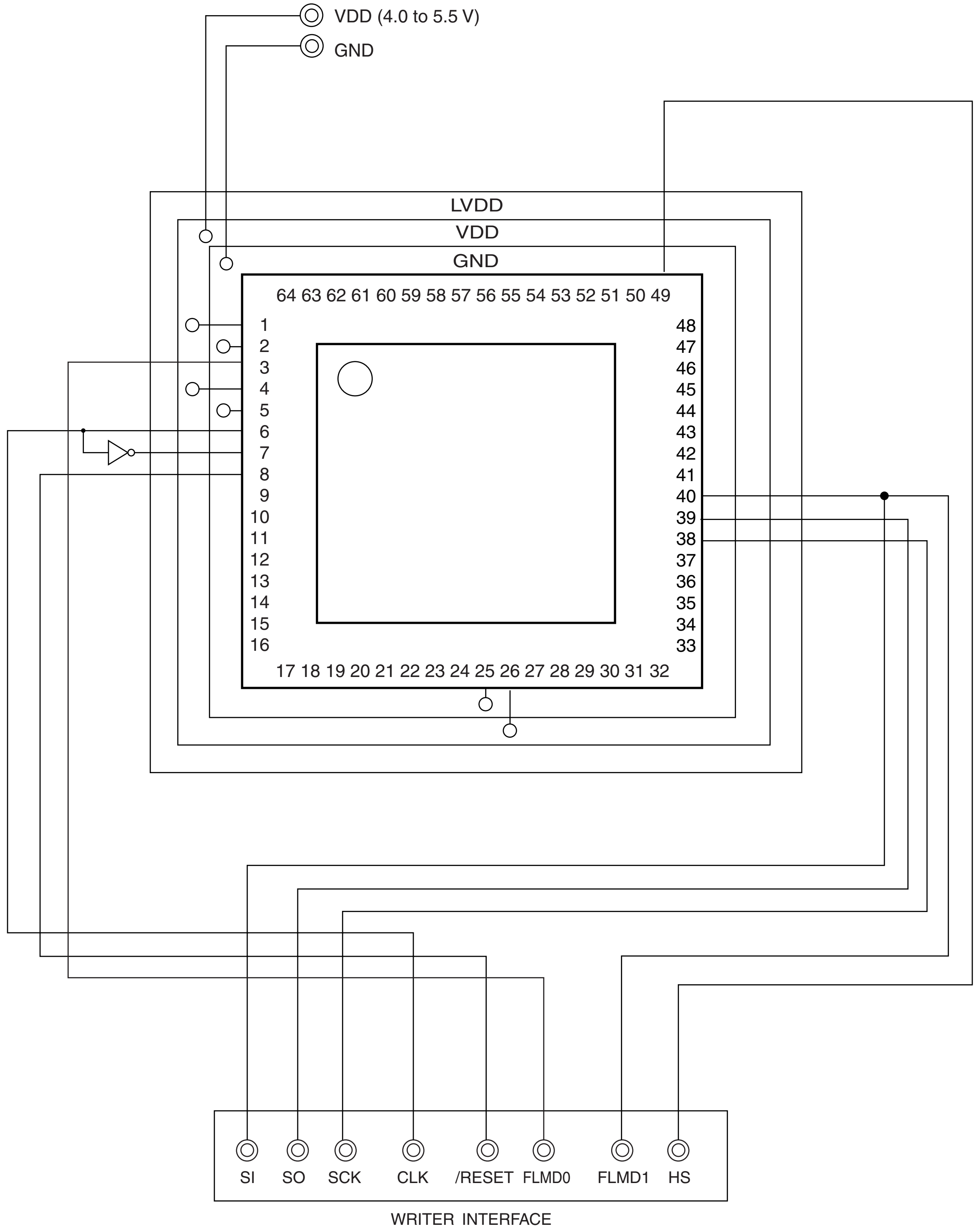
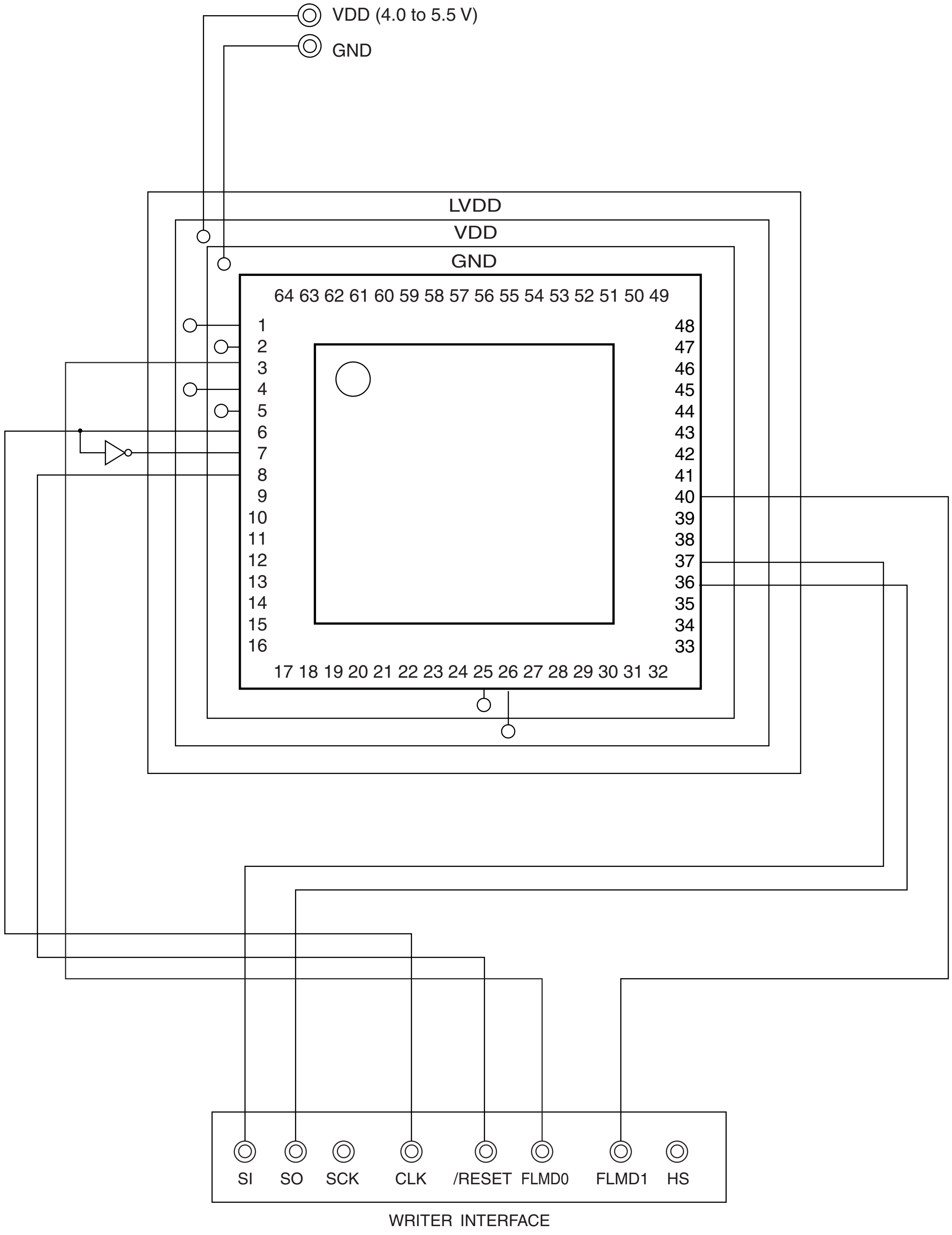


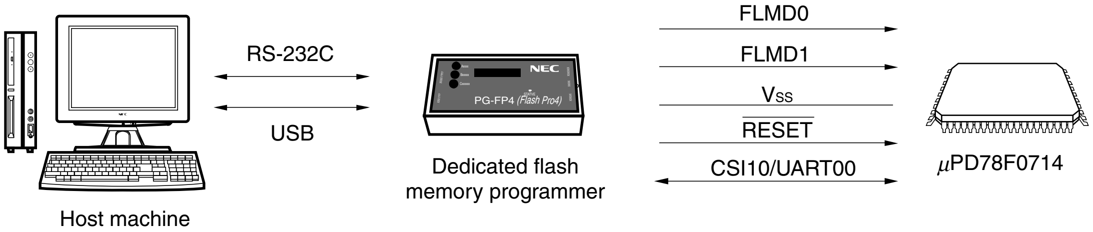
Figure 25-4. Example of Wiring Adapter for Flash Memory Writing in UART (UART0) Mode



25.3 Programming Environment

The environment required for writing a program to the flash memory of the μPD78F0714 is illustrated below.

Figure 25-5. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

CSI10 or UART00 is used to interface between the dedicated flash memory programmer and the μPD78F0714 for manipulation such as writing and erasing. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

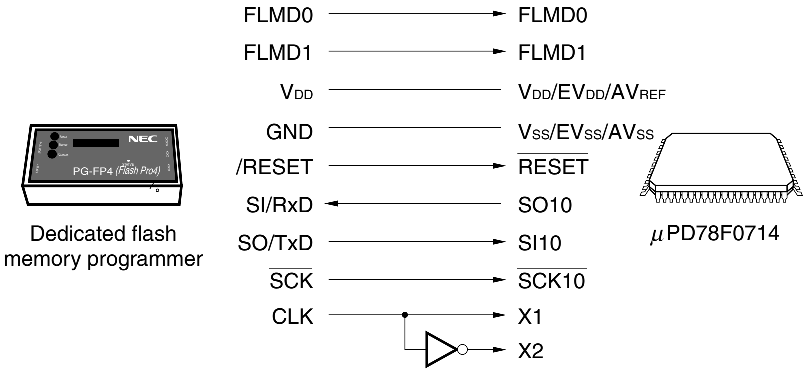
25.4 Communication Mode

Communication between the dedicated flash memory programmer and the μPD78F0714 is established by serial communication via CSI10 or UART00 of the μPD78F0714.

(1) CSI10

Transfer rate: 200 kHz to 2 MHz

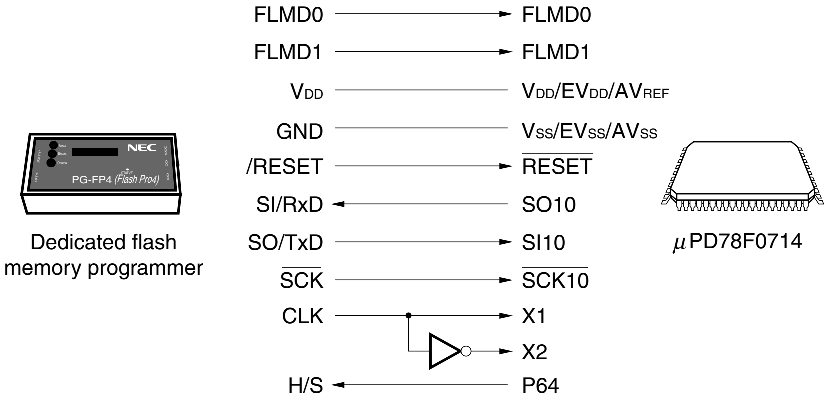
Figure 25-6. Communication with Dedicated Flash Memory Programmer (CSI10)



(2) CSI communication mode supporting handshake

Transfer rate: 200 kHz to 2 MHz

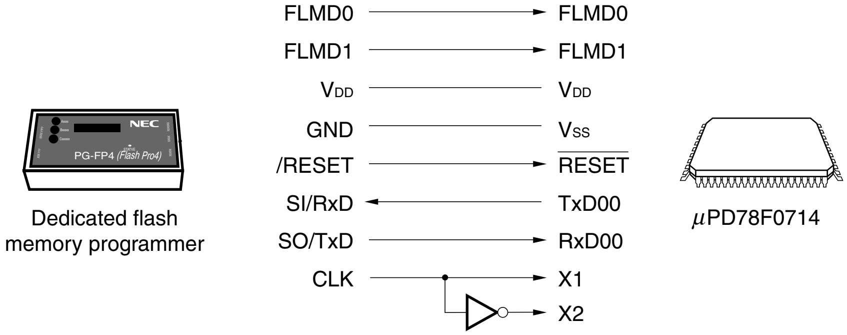
Figure 25-7. Communication with Dedicated Flash Memory Programmer (CSI10 + HS)



(3) UART00

Transfer rate: 4800 to 76800 bps

Figure 25-8. Communication with Dedicated Flash Memory Programmer (UART00)



If Flashpro IV is used as the dedicated flash memory programmer, Flashpro IV generates the following signal for the μPD78F0714. For details, refer to the Flashpro IV Manual.

Table 25-2. Pin Connection

Flashpro IV			μPD78F0714	Connection	
Signal Name	I/O	Pin Function	Pin Name	CSI10	UART00
FLMD0	Output	Mode signal	FLMD0	◎	◎
FLMD1	Output	Mode signal	FLMD1	○	○
V_{DD}	I/O	V_{DD} voltage generation	V_{DD}, EV_{DD}, AV_{REF}	◎	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	◎	◎
CLK	Output	Clock output to μPD78F0714	X1, X2[Note]	○	○
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎
SI/RxD	Input	Receive signal	SO10/TxD00	◎	◎
SO/TxD	Output	Transmit signal	SI10/RxD00	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCK10}}$	◎	×
H/S	Input	Handshake signal	P64	△	×

Note When using the clock out of the flash memory programmer, connect CLK of the programmer to X1, and connect its inverse signal to X2.

Remark ◎: Be sure to connect the pin.
○: The pin does not have to be connected if the signal is generated on the target board.
×: The pin does not have to be connected.
△: In handshake mode

25.5 Processing of Pins on Board

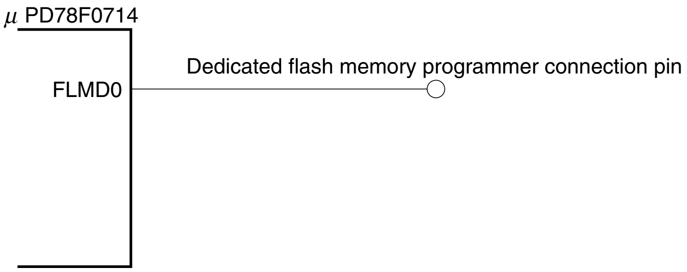
To write the flash memory on-board, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.

When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be processed as described below.

25.5.1 FLMD0 pin

In the normal operation mode, 0 V is input to the FLMD0 pin. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. The following shows an example of the connection of the FLMD0 pin.

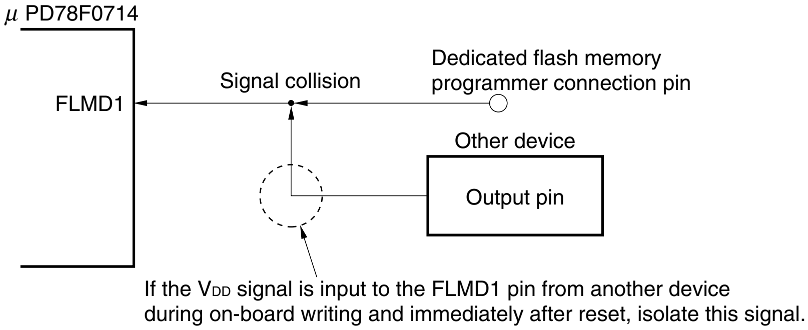
Figure 25-9. FLMD0 Pin Connection Example



25.5.2 FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so FLMD1 must be input to the same as voltage V_{SS}. An FLMD1 pin connection example is shown below.

Figure 25-10. FLMD1 Pin Connection Example


μ PD78F0714
FLMD1
Signal collision
Dedicated flash memory programmer connection pin
Other device
Output pin
If the VDD signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

25.5.3 Serial interface pins

The pins used by each serial interface are listed below.

Table 25-3. Pins Used by Each Serial Interface

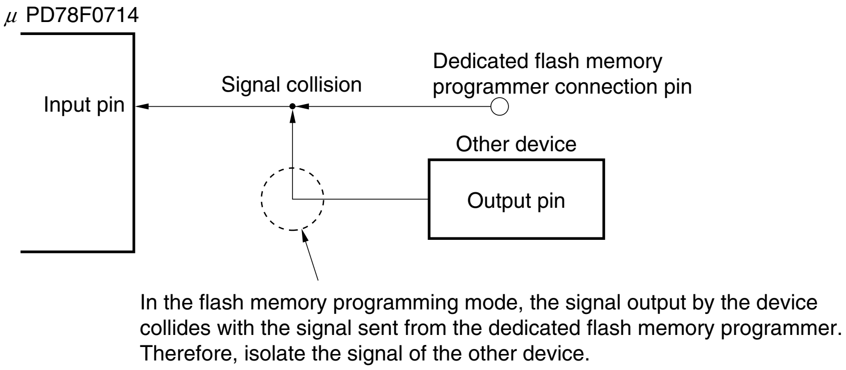
Serial Interface	Pins Used
CSI10	SO10, SI10, $\overline{\text{SCK10}}$
CSI10 + HS	SO10, SI10, $\overline{\text{SCK10}}$, P64
UART00	TxD00, RxD00

To connect the dedicated flash memory programmer to the pins of a serial interface that is connected to another device on the board, care must be exercised so that signals do not collide or that the other device does not malfunction.

(1) Signal collision

If the dedicated flash memory programmer (output) is connected to a pin (input) of a serial interface connected to another device (output), signal collision takes place. To avoid this collision, either isolate the connection with the other device, or make the other device go into an output high-impedance state.

Figure 25-11. Signal Collision (Input Pin of Serial Interface)

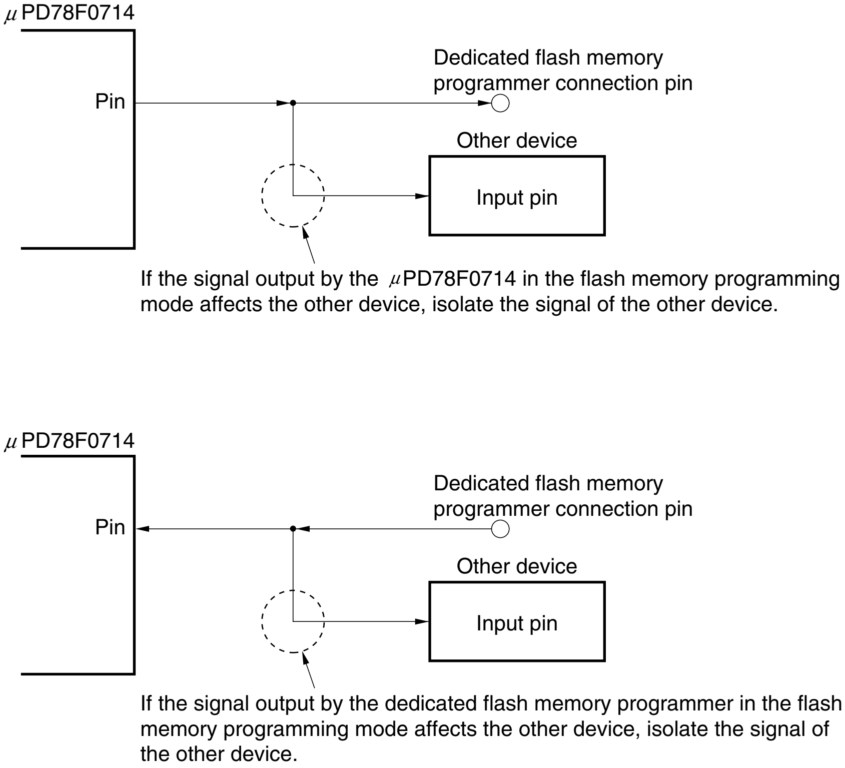


In the flash memory programming mode, the signal output by the device collides with the signal sent from the dedicated flash memory programmer. Therefore, isolate the signal of the other device.

(2) Malfunction of other device

If the dedicated flash memory programmer (output or input) is connected to a pin (input or output) of a serial interface connected to another device (input), a signal may be output to the other device, causing the device to malfunction. To avoid this malfunction, isolate the connection with the other device.

Figure 25-12. Malfunction of Other Device


μPD78F0714
Pin
Dedicated flash memory programmer connection pin
Other device
Input pin

If the signal output by the μPD78F0714 in the flash memory programming mode affects the other device, isolate the signal of the other device.

μPD78F0714
Pin
Dedicated flash memory programmer connection pin
Other device
Input pin

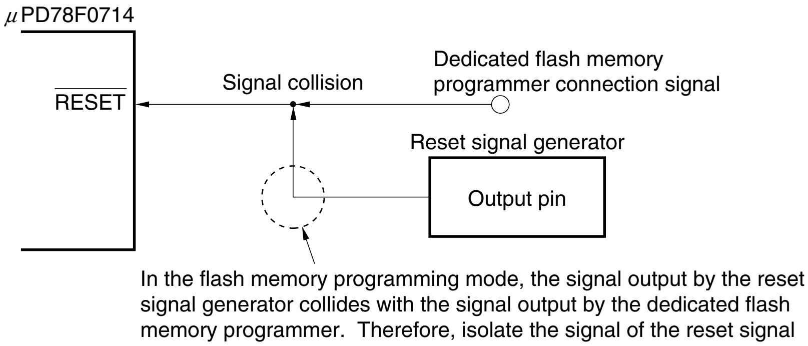
If the signal output by the dedicated flash memory programmer in the flash memory programming mode affects the other device, isolate the signal of the other device.

25.5.4 $\overline{\text{RESET}}$ pin

If the reset signal of the dedicated flash memory programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board, signal collision takes place. To prevent this collision, isolate the connection with the reset signal generator.

If the reset signal is input from the user system while the flash memory programming mode is set, the flash memory will not be correctly programmed. Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 25-13. Signal Collision ($\overline{\text{RESET}}$ Pin)


μPD78F0714
$\overline{\text{RESET}}$
Signal collision
Dedicated flash memory
programmer connection signal
Reset signal generator
Output pin
In the flash memory programming mode, the signal output by the reset signal generator collides with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of the reset signal

25.5.5 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

25.5.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode when using the on-board clock.

To input the operating clock from the programmer, however, connect the clock out of the programmer to X1, and its inverse signal to X2.

25.5.7 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to V_{SS} of the flash memory programmer.

To use the on-board supply voltage, connect in compliance with the normal operation mode.

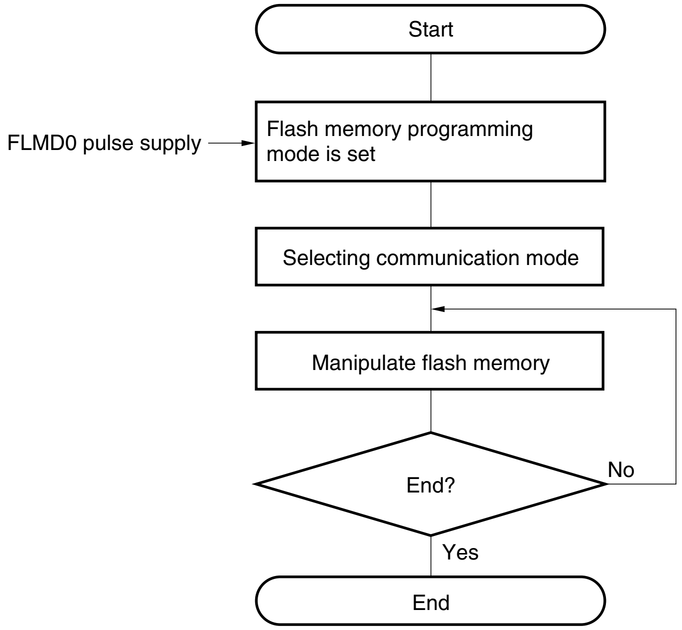
Supply the same other power supplies (EV_{DD}, EV_{SS}, AV_{REF}, and AV_{SS}) as those in the normal operation mode.

25.6 Programming Method

25.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 25-14. Flash Memory Manipulation Procedure


Start
FLMD0 pulse supply
Flash memory programming mode is set
Selecting communication mode
Manipulate flash memory
End?
No
Yes
End

25.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the μPD78F0714 in the flash memory programming mode. To set the mode, set the FLMD0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 25-15. Flash Memory Programming Mode

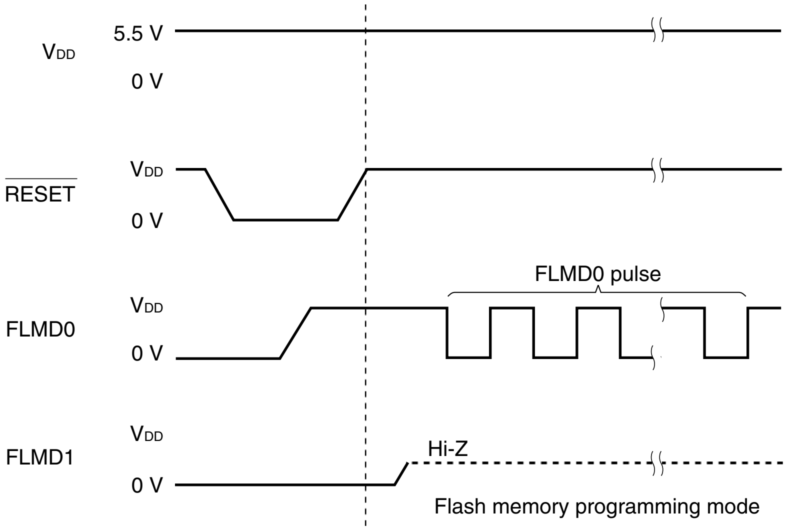


Table 25-4. Relationship of Operation Mode of FLMD0 and FLMD1 Pins

FLMD0	FLMD1	Operation Mode
0	×	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

25.6.3 Selecting communication mode

In the μPD78F0714 a communication mode is selected by inputting pulses (up to 11 pulses) to the FLMD0 pin after the dedicated flash memory programming mode is entered. These FLMD0 pulses are generated by the flash memory programmer.

The following table shows the relationship between the number of pulses and communication modes.

Table 25-5. Communication Modes

Communication Mode	Standard Setting[Note 1]					Pins Used	Number of FLMD0 Pulses
	Port	Speed	On Target	Frequency	Multiply Rate		
<R> UART (UART00)	UART-ch0	9600, 19200, 31250, 38400, 76800, 153600 [Note 2] bps[Note 3]	Optional	5 M to 20 MHz [Note 4]	1.0	TxD00, RxD00	0
3-wire serial I/O (CSI10)	SIO-ch0	200 k to 2 MHz[Note 4]				SO10, SI10, $\overline{SCK10}$	8
3-wire serial I/O with handshake (CSI10 + HS)	SIO-H/S	200 k to 2 MHz[Note 4]				SO10, SI10, $\overline{SCK10}$, P64	11

Notes 1. Selection items for Standard settings on Flashpro IV.

<R> **2.** This cannot be selected, if the peripheral hardware clock frequency is 2.5 MHz or less.

3. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

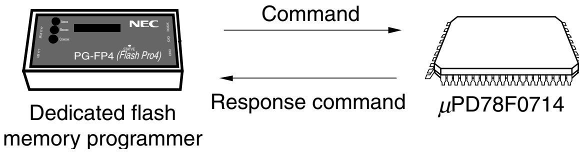
4. The possible setting range differs depending on the voltage. For details, refer to the electrical specifications chapter.

Caution When UART00 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash memory programmer after the FLMD0 pulse has been received.

25.6.4 Communication commands

The μPD78F0714 communicates with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the μPD78F0714 are called commands, and the commands sent from the μPD78F0714 to the dedicated flash memory programmer are called response commands.

Figure 25-16. Communication Commands



The flash memory control commands of the μPD78F0714 are listed in the table below. All these commands are issued from the programmer and the μPD78F0714 perform processing corresponding to the respective commands.

Table 25-6. Flash Memory Control Commands

Classification	Command Name	Function
Verify	Batch verify command	Compares the contents of the entire memory with the input data.
Erase	Batch erase command	Erases the contents of the entire memory.
Blank check	Batch blank check command	Checks the erasure status of the entire memory.
Data write	High-speed write command	Writes data by specifying the write address and number of bytes to be written, and executes a verify check.
	Successive write command	Writes data from the address following that of the high-speed write command executed immediately before, and executes a verify check.
System setting, control	Status read command	Obtains the operation status
	Oscillation frequency setting command	Sets the oscillation frequency
	Erase time setting command	Sets the erase time for batch erase
	Write time setting command	Sets the write time for writing data
	Baud rate setting command	Sets the baud rate when UART is used
	Silicon signature command	Reads the silicon signature information
	Reset command	Escapes from each status

The μPD78F0714 return a response command for the command issued by the dedicated flash memory programmer. The response commands sent from the μPD78F0714 are listed below.

Table 25-7. Response Commands

Command Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

25.7 Flash Memory Programming by Self-Writing

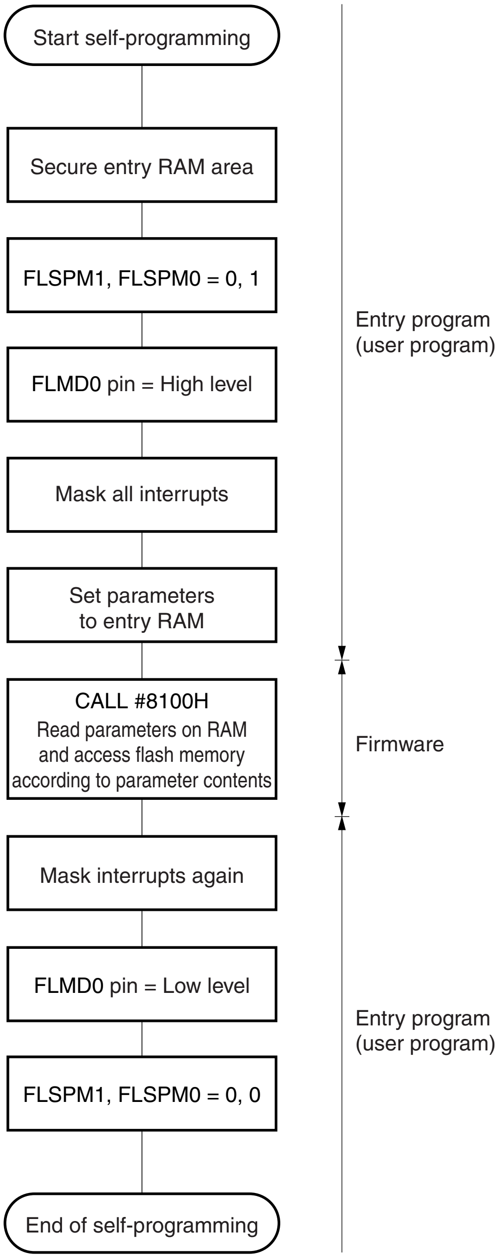
The μPD78F0714 supports a self-programming function that can be used to rewrite the flash memory via a user program, so that the program can be upgraded in the field.

The programming mode is selected by bits 0 and 1 (FLSPM0 and FLSPM1) of the flash programming mode control register (FLPMC).

The procedure of self-programming is illustrated below.

<R> **Remark** For details of the self programming function, refer to a separate document to be published soon (document name: **μPD78F0711, 78F0712, 78F0714 Flash Memory Self Programming User's Manual (U18886E)**).

Figure 25-17. Self-Programming Procedure



25.7.1 Registers used for self-programming function

The following three registers are used for the self-programming function.

- Flash-programming mode control register (FLPMC)
- Flash protect command register (PFCMD)
- Flash status register (PFS)

(1) Flash-programming mode control register (FLPMC)

This register is used to enable or disable writing or erasing of the flash memory and to set the operation mode during self-programming.

The FLPMC can be written only in a specific sequence (see **25.7.1 (2) Flash protect command register**) so that the application system does not stop inadvertently due to malfunction caused by noise or program hang-up.

FLPMC can be set by a 1-bit or 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input sets this register to 0xH[Note].

Note Differs depending on the operation mode.

- User mode: 08H
- On-board mode: 0CH

Figure 25-18. Format of Flash-Programming Mode Control Register (FLPMC)

Address: FFC4H After reset: 0×H[Note 1] R/W[Note 2]

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	0	FWEDIS	FWEPR	FLSPM1	FLSPM0

FWEDIS	Control of flash memory writing/erasing
0	Writing/erasing enabled[Note 3]
1	Writing/erasing disabled

FWEPR	Status of FLMD0 pin
0	Low level
1	High level[Note 3]

FLSPM1[Note 4]	FLSPM0[Note 4]	Selection of operation mode during self-programming
0	0	Normal mode Instructions of flash memory can be fetched from all addresses.
0	1	Self-programming mode A1 Firmware can be called (CALL #8100H).
1	1	Self-programming mode A2 Instructions are fetched from firmware ROM. This mode is set in firmware and cannot be set by the user.
1	0	Setting prohibited

Notes 1. Differs depending on the operation mode.
- User mode: 08H
- On-board mode: 0CH

2. Bit 2 (FWEPR) is read-only.

3. For actual writing/erasing, the FLMPD0 pin must be high (FWEPR = 1), as well as FWEDIS = 0.

FWEDIS	FWEPR	Enable or disable of flash memory writing/erasing
0	1	Writing/erasing enabled
Other than above		Writing/erasing disabled

4. The user ROM (flash memory) or firmware ROM can be selected by FLSPM1 and FLSPM0, and the operation mode set on the application system by the mode pin or the self-programming mode can be selected.

Cautions 1. Be sure to keep FWEDIS at 0 until writing or erasing of the flash memory is completed.

2. Make sure that FWEDIS = 1 in the normal mode.

3. Manipulate FLSPM1 and FLSPM0 after execution branches to the internal RAM. The address of the flash memory is specified by an address signal from the CPU when FLSPM1 = 0 or the set value of the firmware written when FLSPM1 = 1. In the on-board mode, the specifications of FLSPM1 and FLSPM0 are ignored.

(2) Flash protect command register (PFCMD)

If the application system stops inadvertently due to malfunction caused by noise or program hang-up, an operation to write the flash programming mode control register (FLPMC) may have a serious effect on the system. PFCMD is used to protect FLPMC from being written, so that the application system does not stop inadvertently.

Writing FLMPC is enabled only when a write operation is performed in the following specific sequence.

<1> Write a specific value to PFCMD (PFCMD = A5H)
<2> Write the value to be set to FLPMC (writing in this step is invalid)
<3> Write the inverted value of the value to be set to FLPMC (writing in this step is invalid)
<4> Write the value to be set to FLPMC (writing in this step is valid)

This rewrites the value of the register, so that the register cannot be written illegally.
Occurrence of an illegal store operation can be checked by bit 0 (FPRERR) of the flash status register (PFS).
A5H must be written to PFCMD each time the value of FLPMC is changed.
PFCMD can be set by an 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input makes this register undefined.

Figure 25-19. Format of Flash Protect Command Register (PFCMD)

Address: FFC0H After reset: Undefined W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) Flash status register (PFS)

If data is not written to the flash programming mode control register (FLPMC), which is protected, in the correct sequence (writing the flash protect command register (PFCMD)), FLPMC is not written and a protection error occurs. If this happens, bit 0 of PFS (FPRERR) is set to 1.
This bit is a cumulative flag. After checking FPRERR, clear it by writing 0 to it.
PFS can be set by a 1-bit or 8-bit memory manipulation instruction.
$\overline{\text{RESET}}$ input clears this register to 00H.

Figure 25-20. Format of Flash Status Register (PFS)

Address: FFC2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PFS	0	0	0	0	0	0	0	FPRERR

The operating conditions of the FPRERR flag are as follows.

<Setting conditions>

- If PFCMD is written when the store instruction operation recently performed on a peripheral register is not to write a specific value (A5H) to PFCMD
- If the first store instruction operation after <1> is on a peripheral register other than FLPMC
- If the first store instruction operation after <2> is on a peripheral register other than FLPMC
- If a value other than the inverted value of the value to be set to FLPMC is written by the first store instruction after <2>
- If the first store instruction operation after <3> is on a peripheral register other than FLPMC
- If a value other than the value to be set to FLPMC (value written in <2>) is written by the first store instruction after <3>

Remark The numbers in angle brackets above correspond to the those in **(2) Flash protect command register (PFCMD)**.

<Reset conditions>

- If 0 is written to the FPRERR flag
- If $\overline{\text{RESET}}$ is input

<Example of description in specific sequence

To write 05H to FLPMC

```
MOV    PFCMD, #0A5H     ; Writes A5H to PFCMD.
MOV    FLPMC, #05H      ; Writes 05H to FLPMC.
MOV    FLPMC, #0FAH     ; Writes 0FAH (inverted value of 05H) to FLPMC.
MOV    FLPMC, #05H      ; Writes 05H to FLPMC.
```

25.8 Boot Swap Function

The μPD78F0714 has a boot swap function.

Even if a momentary power failure occurs for some reason while the boot area is being rewritten by self-programming and the program in the boot area is lost, the boot swap function can execute the program correctly after re-application of power, reset, and start.

25.8.1 Outline of boot swap function

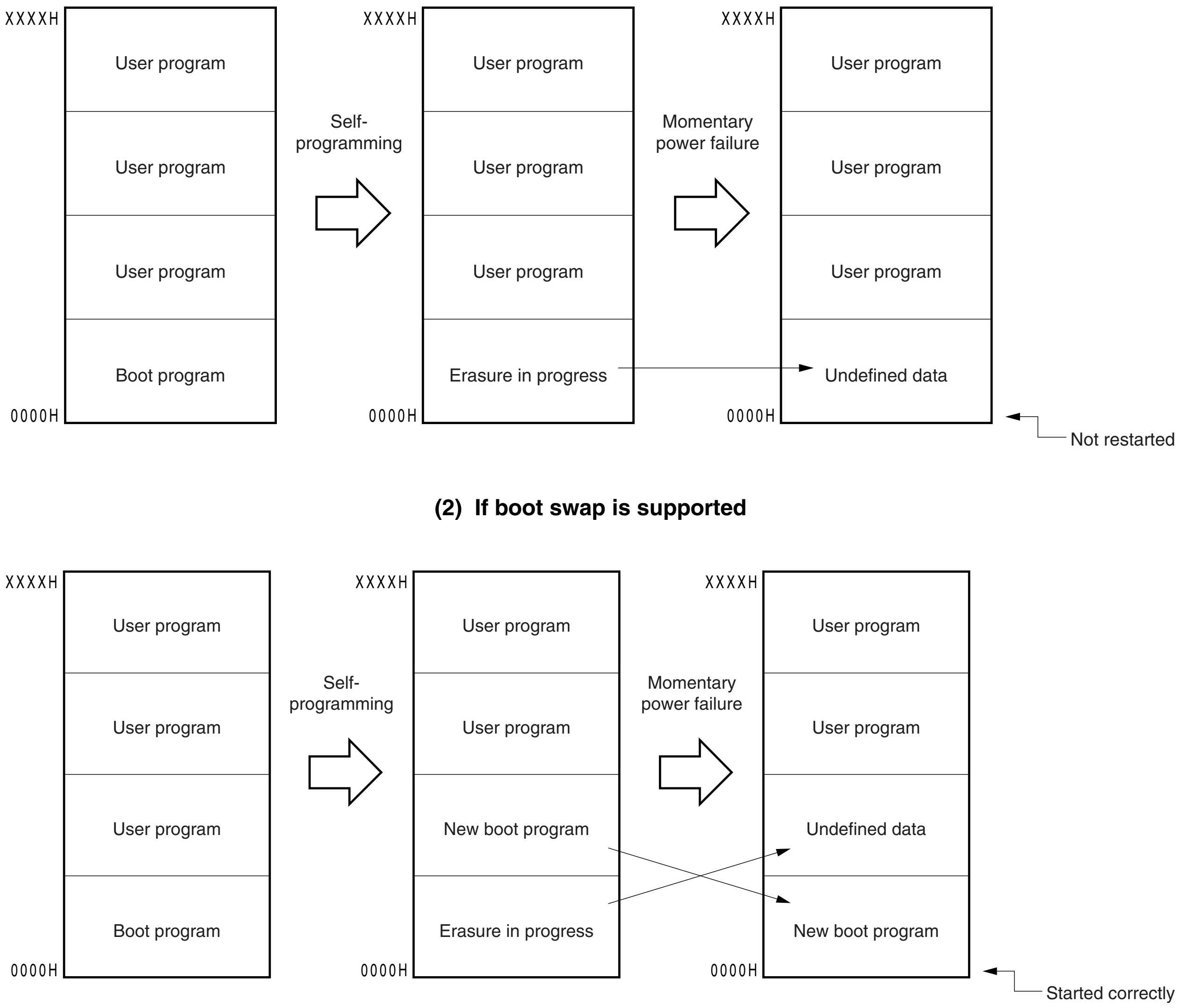
Before erasing the boot program area by self-programming, write a new boot program to the block to be swapped, and also set the boot flag[Note]. Even if a momentary power failure occurs, the address is swapped when the system is reset and started next time. Consequently, the above area to be swapped is used as a boot area, and the program is executed correctly. **Figure 25-21** shows an image of the boot swap function.

Note The boot flag is in the flash memory control firmware of the μPD78F0714.

Figure 25-21. Image of Boot Swap Function

(1) If boot swap is not supported

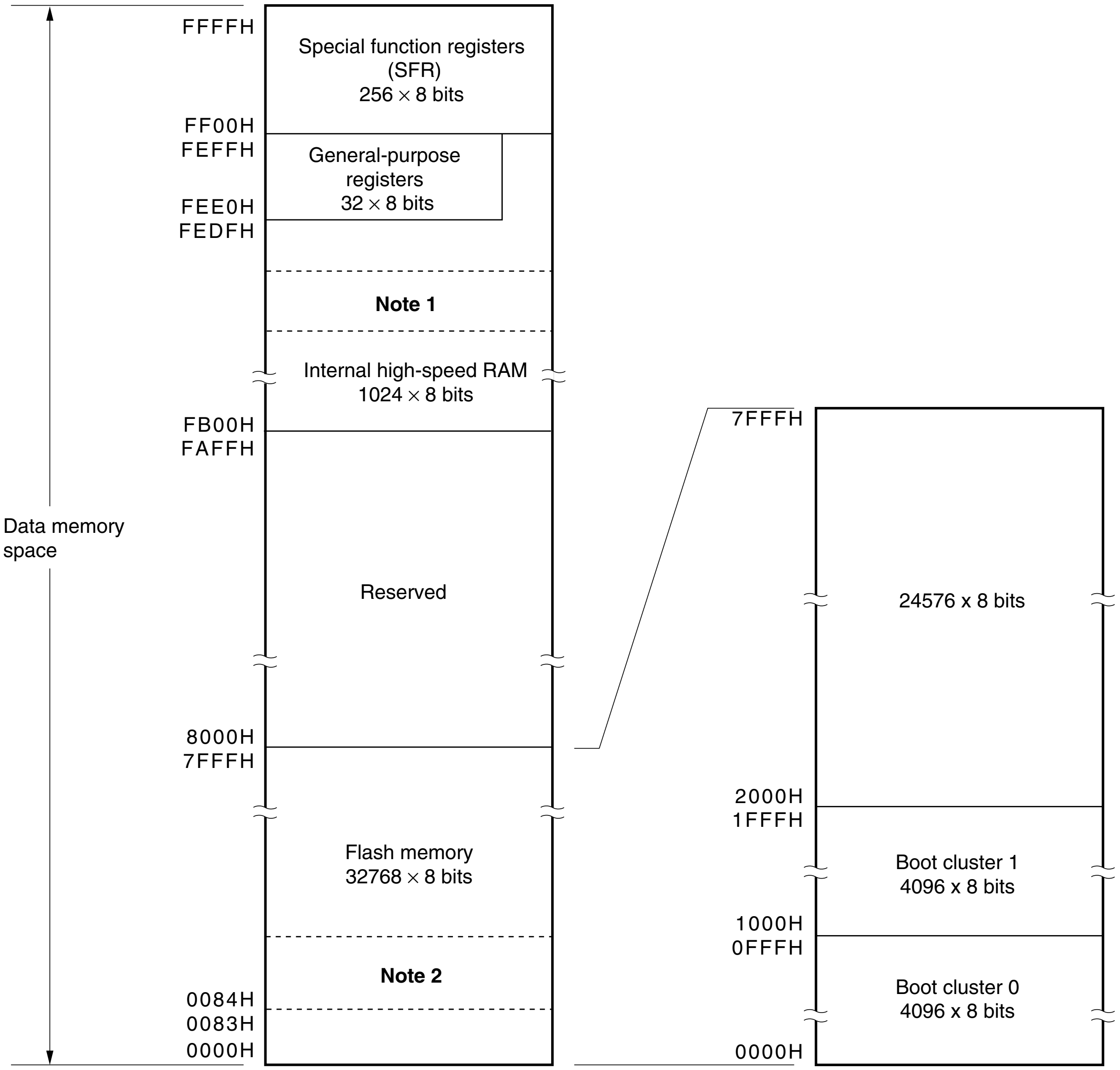
(2) If boot swap is supported



25.8.2 Memory map and boot area

Figure 25-22 shows the memory map and boot area. The boot program area of the μPD78F0714 is in 4 KB units. When boot swap is executed, boot cluster 0 and boot cluster 1 in the figure are exchanged.

Figure 25-22. Memory Map and Boot Area


FFFFH
Special function registers (SFR) 256 × 8 bits
FF00H
FEFFH
General-purpose registers 32 × 8 bits
FEE0H
FEDFH
Note 1
Internal high-speed RAM 1024 × 8 bits
FB00H
FAFFH
Data memory space
Reserved
8000H
7FFFH
Flash memory 32768 × 8 bits
Note 2
0084H
0083H
0000H
7FFFH
24576 x 8 bits
2000H
1FFFH
Boot cluster 1 4096 x 8 bits
1000H
0FFFH
Boot cluster 0 4096 x 8 bits
0000H

Notes 1. This area of the product of 9 byte (schedule) can be used during on-chip debugging because it is used as a backup area of user data during communication.

2. This area of the product cannot be used during on-chip debugging because it is used as a communication command area (256 bytes to 1 KB).

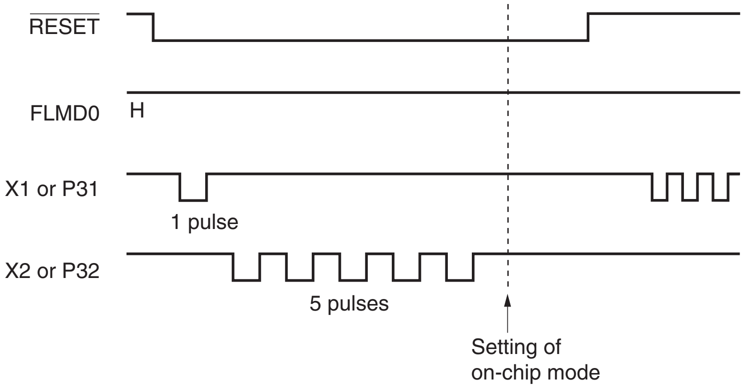
<R> The μPD78F0714 uses the V_{DD}, FLMD0, $\overline{RESET}$, X1 (or P31), X2 (or P32), and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-78K0MINI or QB-MINI2) for on-chip debugging. Whether X1 and P31, or X2 and P32 are used can be selected.

Cautions 1. Be sure to pull down P31 after reset to prevent malfunction.

2. When using P31 for the on-chip debug function, it is recommended not to use P31 for any purpose other than on-chip debugging.

<R> **Remark** For details of the on-chip debug function, refer to **QB-78K0MINI User's Manual (U17029E)** or **QB-MINI2 User's Manual (U18371E)**.

Figure 26-1. Timing Chart of Setting On-Chip Debug Mode



CHAPTER 27 INSTRUCTION SET

This chapter lists each instruction set of the μPD78F0714 in table form. For details of each operation and operation code, refer to the separate document **78K/0 Series Instructions User's Manual (U12326E)**.

27.1 Conventions Used in Operation List

27.1.1 Operand identifiers and specification methods

Operands are written in the "Operand" column of each instruction in accordance with the specification method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more methods, select one of them. Uppercase letters and the symbols #, !, $ and [ ] are keywords and must be written as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: Absolute address specification
- $: Relative address specification
- [ ]: Indirect address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to write the #, !, $, and [ ] symbols.

For operand register identifiers r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for specification.

Table 27-1. Operand Identifiers and Specification Methods

Identifier	Specification Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special function register symbol[Note] Special function register symbol (16-bit manipulatable register even addresses only)[Note]
saddr saddrp	FE20H to FF1FH Immediate data or labels FE20H to FF1EH Immediate data or labels (even address only)
addr16 addr11 addr5	0000H to FFFFH Immediate data or labels (Only even addresses for 16-bit data transfer instructions) 0800H to 0FFFH Immediate data or labels 0040H to 007EH Immediate data or labels (even address only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Addresses from FFD0H to FFDFH cannot be accessed with these operands.

Remark For special function register symbols, see **Table 3-3 Special Function Register List**.

27.1.2 Description of operation column

A:	A register; 8-bit accumulator
X:	X register
B:	B register
C:	C register
D:	D register
E:	E register
H:	H register
L:	L register
AX:	AX register pair; 16-bit accumulator
BC:	BC register pair
DE:	DE register pair
HL:	HL register pair
PC:	Program counter
SP:	Stack pointer
PSW:	Program status word
CY:	Carry flag
AC:	Auxiliary carry flag
Z:	Zero flag
RBS:	Register bank select flag
IE:	Interrupt request enable flag
NMIS:	Non-maskable interrupt servicing flag
( ):	Memory contents indicated by address or register contents in parentheses
X_H, X_L:	Higher 8 bits and lower 8 bits of 16-bit register
$\wedge$:	Logical product (AND)
$\vee$:	Logical sum (OR)
$\veebar$:	Exclusive logical sum (exclusive OR)
$\overline{\ \ }$:	Inverted data
addr16:	16-bit immediate data or label
addr11:	Immediate data or label
addr5:	Immediate data or label (even address only)
jdisp8:	Signed 8-bit data (displacement value)

27.1.3 Description of flag operation column

(Blank):	Not affected
0:	Cleared to 0
1:	Set to 1
×:	Set/cleared according to the result
R:	Previously saved value is restored

27.2 Operation List

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	**MOV**	r, #byte	2	4	–	r ← byte	
		saddr, #byte	3	6	7	(saddr) ← byte	
		sfr, #byte	3	–	7	sfr ← byte	
		A, r Note 3	1	2	–	A ← r	
		r, A Note 3	1	2	–	r ← A	
		A, saddr	2	4	5	A ← (saddr)	
		saddr, A	2	4	5	(saddr) ← A	
		A, sfr	2	–	5	A ← sfr	
		sfr, A	2	–	5	sfr ← A	
		A, !addr16	3	8	9	A ← (addr16)	
		!addr16, A	3	8	9	(addr16) ← A	
		PSW, #byte	3	–	7	PSW ← byte	× × ×
		A, PSW	2	–	5	A ← PSW	
		PSW, A	2	–	5	PSW ← A	× × ×
		A, [DE]	1	4	5	A ← (DE)	
		[DE], A	1	4	5	(DE) ← A	
		A, [HL]	1	4	5	A ← (HL)	
		[HL], A	1	4	5	(HL) ← A	
		A, [HL + byte]	2	8	9	A ← (HL + byte)	
		[HL + byte], A	2	8	9	(HL + byte) ← A	
		A, [HL + B]	1	6	7	A ← (HL + B)	
		[HL + B], A	1	6	7	(HL + B) ← A	
		A, [HL + C]	1	6	7	A ← (HL + C)	
		[HL + C], A	1	6	7	(HL + C) ← A	
	XCH	A, r Note 3	1	2	–	A ↔ r	
		A, saddr	2	4	6	A ↔ (saddr)	
		A, sfr	2	–	6	A ↔ (sfr)	
		A, !addr16	3	8	10	A ↔ (addr16)	
		A, [DE]	1	4	6	A ↔ (DE)	
		A, [HL]	1	4	6	A ↔ (HL)	
		A, [HL + byte]	2	8	10	A ↔ (HL + byte)	
		A, [HL + B]	2	8	10	A ↔ (HL + B)	
		A, [HL + C]	2	8	10	A ↔ (HL + C)	

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	**MOVW**	rp, #word	3	6	–	rp ← word			
		saddrp, #word	4	8	10	(saddrp) ← word			
		sfrp, #word	4	–	10	sfrp ← word			
		AX, saddrp	2	6	8	AX ← (saddrp)			
		saddrp, AX	2	6	8	(saddrp) ← AX			
		AX, sfrp	2	–	8	AX ← sfrp			
		sfrp, AX	2	–	8	sfrp ← AX			
		AX, rp Note 3	1	4	–	AX ← rp			
		rp, AX Note 3	1	4	–	rp ← AX			
		AX, !addr16	3	10	12	AX ← (addr16)			
		!addr16, AX	3	10	12	(addr16) ← AX			
	XCHW	AX, rp Note 3	1	4	–	AX ↔ rp			
8-bit operation	**ADD**	A, #byte	2	4	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	4	–	A, CY ← A + r	×	×	×
		r, A	2	4	–	r, CY ← r + A	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C)	×	×	×
	ADDC	A, #byte	2	4	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 4	2	4	–	A, CY ← A + r + CY	×	×	×
		r, A	2	4	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	4	5	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A + (HL + C) + CY	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Only when rp = BC, DE or HL

4. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**SUB**	A, #byte	2	4	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r	×	×	×
		r, A	2	4	–	r, CY ← r – A	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C)	×	×	×
	SUBC	A, #byte	2	4	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	6	8	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	4	–	A, CY ← A – r – CY	×	×	×
		r, A	2	4	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	4	5	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	8	9	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	4	5	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	8	9	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	8	9	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	8	9	A, CY ← A – (HL + C) – CY	×	×	×
	AND	A, #byte	2	4	–	A ← A ∧ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	4	–	A ← A ∧ r	×		
		r, A	2	4	–	r ← r ∧ A	×		
		A, saddr	2	4	5	A ← A ∧ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∧ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∧ (HL + C)	×		

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	**OR**	A, #byte	2	4	–	A ← A ∨ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	4	–	A ← A ∨ r	×		
		r, A	2	4	–	r ← r ∨ A	×		
		A, saddr	2	4	5	A ← A ∨ (saddr)	×		
		A, !addr16	3	8	9	A ← A ∨ (addr16)	×		
		A, [HL]	1	4	5	A ← A ∨ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ∨ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ∨ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ∨ (HL + C)	×		
	XOR	A, #byte	2	4	–	A ← A ⊻ byte	×		
		saddr, #byte	3	6	8	(saddr) ← (saddr) ⊻ byte	×		
		A, r Note 3	2	4	–	A ← A ⊻ r	×		
		r, A	2	4	–	r ← r ⊻ A	×		
		A, saddr	2	4	5	A ← A ⊻ (saddr)	×		
		A, !addr16	3	8	9	A ← A ⊻ (addr16)	×		
		A, [HL]	1	4	5	A ← A ⊻ (HL)	×		
		A, [HL + byte]	2	8	9	A ← A ⊻ (HL + byte)	×		
		A, [HL + B]	2	8	9	A ← A ⊻ (HL + B)	×		
		A, [HL + C]	2	8	9	A ← A ⊻ (HL + C)	×		
	CMP	A, #byte	2	4	–	A – byte	×	×	×
		saddr, #byte	3	6	8	(saddr) – byte	×	×	×
		A, r Note 3	2	4	–	A – r	×	×	×
		r, A	2	4	–	r – A	×	×	×
		A, saddr	2	4	5	A – (saddr)	×	×	×
		A, !addr16	3	8	9	A – (addr16)	×	×	×
		A, [HL]	1	4	5	A – (HL)	×	×	×
		A, [HL + byte]	2	8	9	A – (HL + byte)	×	×	×
		A, [HL + B]	2	8	9	A – (HL + B)	×	×	×
		A, [HL + C]	2	8	9	A – (HL + C)	×	×	×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

3. Except “r = A”

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	**ADDW**	AX, #word	3	6	–	AX, CY ← AX + word	×	×	×
	SUBW	AX, #word	3	6	–	AX, CY ← AX – word	×	×	×
	CMPW	AX, #word	3	6	–	AX – word	×	×	×
Multiply/ divide	**MULU**	X	2	16	–	AX ← A × X			
	DIVUW	C	2	25	–	AX (Quotient), C (Remainder) ← AX ÷ C			
Increment/ decrement	**INC**	r	1	2	–	r ← r + 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) + 1	×	×	
	DEC	r	1	2	–	r ← r – 1	×	×	
		saddr	2	4	6	(saddr) ← (saddr) – 1	×	×	
	INCW	rp	1	4	–	rp ← rp + 1			
	DECW	rp	1	4	–	rp ← rp – 1			
Rotate	**ROR**	A, 1	1	2	–	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1 time			×
	ROL	A, 1	1	2	–	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1 time			×
	RORC	A, 1	1	2	–	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1 time			×
	ROLC	A, 1	1	2	–	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1 time			×
	ROR4	[HL]	2	10	12	A_{3-0} ← $(HL)_{3-0}$, $(HL)_{7-4}$ ← A_{3-0}, $(HL)_{3-0}$ ← $(HL)_{7-4}$			
	ROL4	[HL]	2	10	12	A_{3-0} ← $(HL)_{7-4}$, $(HL)_{3-0}$ ← A_{3-0}, $(HL)_{7-4}$ ← $(HL)_{3-0}$			
BCD adjustment	**ADJBA**		2	4	–	Decimal Adjust Accumulator after Addition	×	×	×
	ADJBS		2	4	–	Decimal Adjust Accumulator after Subtract	×	×	×
Bit manipulate	**MOV1**	CY, saddr.bit	3	6	7	CY ← (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← sfr.bit			×
		CY, A.bit	2	4	–	CY ← A.bit			×
		CY, PSW.bit	3	–	7	CY ← PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← (HL).bit			×
		saddr.bit, CY	3	6	8	(saddr.bit) ← CY			
		sfr.bit, CY	3	–	8	sfr.bit ← CY			
		A.bit, CY	2	4	–	A.bit ← CY			
		PSW.bit, CY	3	–	8	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	6	8	(HL).bit ← CY			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	**AND1**	CY, saddr.bit	3	6	7	CY ← CY ∧ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∧ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∧ (HL).bit			×
	OR1	CY, saddr.bit	3	6	7	CY ← CY ∨ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ∨ (HL).bit			×
	XOR1	CY, saddr.bit	3	6	7	CY ← CY ⊻ (saddr.bit)			×
		CY, sfr.bit	3	–	7	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	4	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	–	7	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	6	7	CY ← CY ⊻ (HL).bit			×
	SET1	saddr.bit	2	4	6	(saddr.bit) ← 1			
		sfr.bit	3	–	8	sfr.bit ← 1			
		A.bit	2	4	–	A.bit ← 1			
		PSW.bit	2	–	6	PSW.bit ← 1	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 1			
	CLR1	saddr.bit	2	4	6	(saddr.bit) ← 0			
		sfr.bit	3	–	8	sfr.bit ← 0			
		A.bit	2	4	–	A.bit ← 0			
		PSW.bit	2	–	6	PSW.bit ← 0	×	×	×
		[HL].bit	2	6	8	(HL).bit ← 0			
	SET1	CY	1	2	–	CY ← 1			1
	CLR1	CY	1	2	–	CY ← 0			0
	NOT1	CY	1	2	–	CY ← $\overline{\text{CY}}$			×

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	**CALL**	!addr16	3	7	–	$(SP - 1) \leftarrow (PC + 3)_H$, $(SP - 2) \leftarrow (PC + 3)_L$, $PC \leftarrow addr16$, $SP \leftarrow SP - 2$			
	CALLF	!addr11	2	5	–	$(SP - 1) \leftarrow (PC + 2)_H$, $(SP - 2) \leftarrow (PC + 2)_L$, $PC_{15-11} \leftarrow 00001$, $PC_{10-0} \leftarrow addr11$, $SP \leftarrow SP - 2$			
<R>	**CALLT**	[addr5]	1	6	–	$(SP - 1) \leftarrow (PC + 1)_H$, $(SP - 2) \leftarrow (PC + 1)_L$, $PC_H \leftarrow (addr5 + 1)$, $PC_L \leftarrow (addr5)$, $SP \leftarrow SP - 2$			
	BRK		1	6	–	$(SP - 1) \leftarrow PSW$, $(SP - 2) \leftarrow (PC + 1)_H$, $(SP - 3) \leftarrow (PC + 1)_L$, $PC_H \leftarrow (003FH)$, $PC_L \leftarrow (003EH)$, $SP \leftarrow SP - 3$, $IE \leftarrow 0$			
	RET		1	6	–	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
	RETI		1	6	–	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP + 2)$, $SP \leftarrow SP + 3$	R	R	R
	RETB		1	6	–	$PC_H \leftarrow (SP + 1)$, $PC_L \leftarrow (SP)$, $PSW \leftarrow (SP + 2)$, $SP \leftarrow SP + 3$	R	R	R
Stack manipulate	**PUSH**	PSW	1	2	–	$(SP - 1) \leftarrow PSW$, $SP \leftarrow SP - 1$			
		rp	1	4	–	$(SP - 1) \leftarrow rp_H$, $(SP - 2) \leftarrow rp_L$, $SP \leftarrow SP - 2$			
	POP	PSW	1	2	–	$PSW \leftarrow (SP)$, $SP \leftarrow SP + 1$	R	R	R
		rp	1	4	–	$rp_H \leftarrow (SP + 1)$, $rp_L \leftarrow (SP)$, $SP \leftarrow SP + 2$			
	MOVW	SP, #word	4	–	10	$SP \leftarrow word$			
		SP, AX	2	–	8	$SP \leftarrow AX$			
		AX, SP	2	–	8	$AX \leftarrow SP$			
Unconditional branch	**BR**	!addr16	3	6	–	$PC \leftarrow addr16$			
		$addr16	2	6	–	$PC \leftarrow PC + 2 + jdisp8$			
		AX	2	8	–	$PC_H \leftarrow A$, $PC_L \leftarrow X$			
Conditional branch	**BC**	$addr16	2	6	–	$PC \leftarrow PC + 2 + jdisp8$ if $CY = 1$			
	BNC	$addr16	2	6	–	$PC \leftarrow PC + 2 + jdisp8$ if $CY = 0$			
	BZ	$addr16	2	6	–	$PC \leftarrow PC + 2 + jdisp8$ if $Z = 1$			
	BNZ	$addr16	2	6	–	$PC \leftarrow PC + 2 + jdisp8$ if $Z = 0$			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	**BT**	saddr.bit, $addr16	3	8	9	PC ← PC + 3 + jdisp8 if (saddr.bit) = 1			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr16	3	–	9	PC ← PC + 3 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
	BF	saddr.bit, $addr16	4	10	11	PC ← PC + 4 + jdisp8 if (saddr.bit) = 0			
		sfr.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr16	4	–	11	PC ← PC + 4 + jdisp8 if PSW. bit = 0			
		[HL].bit, $addr16	3	10	11	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
	BTCLR	saddr.bit, $addr16	4	10	12	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)			
		sfr.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr16	3	8	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr16	4	–	12	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr16	3	10	12	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
	DBNZ	B, $addr16	2	6	–	B ← B – 1, then PC ← PC + 2 + jdisp8 if B ≠ 0			
		C, $addr16	2	6	–	C ← C –1, then PC ← PC + 2 + jdisp8 if C ≠ 0			
		saddr, $addr16	3	8	10	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0			
CPU control	**SEL**	RBn	2	4	–	RBS1, 0 ← n			
	NOP		1	2	–	No Operation			
	EI		2	–	6	IE ← 1 (Enable Interrupt)			
	DI		2	–	6	IE ← 0 (Disable Interrupt)			
	HALT		2	6	–	Set HALT Mode			
	STOP		2	6	–	Set STOP Mode			

Notes 1. When the internal high-speed RAM area is accessed or for an instruction with no data access

2. When an area except the internal high-speed RAM area is accessed

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the processor clock control register (PCC).

2. This clock cycle applies to the internal ROM program.

27.3 Instructions Listed by Addressing Type

(1) 8-bit instructions

MOV, XCH, ADD, ADDC, SUB, SUBC, AND, OR, XOR, CMP, MULU, DIVUW, INC, DEC, ROR, ROL, RORC, ROLC, ROR4, ROL4, PUSH, POP, DBNZ

Second Operand First Operand	#byte	A	r[Note]	sfr	saddr	!addr16	PSW	[DE]	[HL]	[HL + byte] [HL + B] [HL + C]	$addr16	1	None
A	ADD ADDC SUB SUBC AND OR XOR CMP		MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV	MOV XCH	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP	MOV XCH ADD ADDC SUB SUBC AND OR XOR CMP		ROR ROL RORC ROLC	
r	MOV	MOV ADD ADDC SUB SUBC AND OR XOR CMP											INC DEC
B, C											DBNZ		
sfr	MOV	MOV											
saddr	MOV ADD ADDC SUB SUBC AND OR XOR CMP	MOV									DBNZ		INC DEC
!addr16		MOV											
PSW	MOV	MOV											PUSH POP
[DE]		MOV											
[HL]		MOV											ROR4 ROL4
[HL + byte] [HL + B] [HL + C]		MOV											
X													MULU
C													DIVUW

Note Except r = A

(2) 16-bit instructions

MOVW, XCHW, ADDW, SUBW, CMPW, PUSH, POP, INCW, DECW

Second Operand First Operand	#word	AX	rp[Note]	sfrp	saddrp	!addr16	SP	None
AX	ADDW SUBW CMPW		MOVW XCHW	MOVW	MOVW	MOVW	MOVW	
rp	MOVW	MOVW[Note]						INCW DECW PUSH POP
sfrp	MOVW	MOVW						
saddrp	MOVW	MOVW						
!addr16		MOVW						
SP	MOVW	MOVW						

Note Only when rp = BC, DE, HL

(3) Bit manipulation instructions

MOV1, AND1, OR1, XOR1, SET1, CLR1, NOT1, BT, BF, BTCLR

Second Operand First Operand	A.bit	sfr.bit	saddr.bit	PSW.bit	[HL].bit	CY	$addr16	None
A.bit						MOV1	BT BF BTCLR	SET1 CLR1
sfr.bit						MOV1	BT BF BTCLR	SET1 CLR1
saddr.bit						MOV1	BT BF BTCLR	SET1 CLR1
PSW.bit						MOV1	BT BF BTCLR	SET1 CLR1
[HL].bit						MOV1	BT BF BTCLR	SET1 CLR1
CY	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1	MOV1 AND1 OR1 XOR1			SET1 CLR1 NOT1

(4) Call instructions/branch instructions

CALL, CALLF, CALLT, BR, BC, BNC, BZ, BNZ, BT, BF, BTCLR, DBNZ

Second Operand First Operand	AX	!addr16	!addr11	[addr5]	$addr16
Basic instruction	BR	CALL BR	CALLF	CALLT	BR BC BNC BZ BNZ
Compound instruction					BT BF BTCLR DBNZ

(5) Other instructions

ADJBA, ADJBS, BRK, RET, RETI, RETB, SEL, NOP, EI, DI, HALT, STOP

CHAPTER 28 ELECTRICAL SPECIFICATIONS

<R> **Caution The μPD78F0714 has an on-chip debug function. Do not use this product for mass production because its reliability cannot be guaranteed after the on-chip debug function has been used, given the issue of the number of times the flash memory can be rewritten. NEC Electronics does not accept complaints concerning this product.**

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			–0.3 to +6.5	V
	EV_{DD}			–0.3 to +6.5	V
	V_{SS}			–0.3 to +0.3	V
	EV_{SS}			–0.3 to +0.3	V
	AV_{REF}			–0.3 to V_{DD} + 0.3[Note]	V
	AV_{SS}			–0.3 to +0.3	V
Input voltage	V_I	P00 to P03, P10 to P17, P20 to P27, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P73, X1, X2, $\overline{\text{RESET}}$		–0.3 to V_{DD} + 0.3[Note]	V
Output voltage	V_O			–0.3 to V_{DD} + 0.3[Note]	V
Analog input voltage	V_{AN}			AV_{SS} – 0.3 to AV_{REF} + 0.3[Note] and –0.3 to V_{DD} + 0.3[Note]	V
Output current, high	I_{OH}	Per pin		–10	mA
		Total of all pins –60 mA	P00 to P03, P30 to P33, P50 to P57	–30	mA
			P10 to P17, P40 to P47, P64 to P67, P70 to P73, TW0TO0 to TW0TO5	–30	mA

Note Must be 6.5 V or lower.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

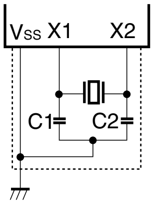
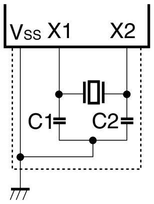
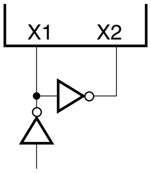
Parameter	Symbol	Conditions		Ratings	Unit
Output current, low	I_{OL}	Per pin	P00 to P03, P10 to P17, P30 to P33, P40 to P47, P64 to P67, P70 to P73	20	mA
			P50 to P57, TW0TO0 to TW0TO5	30	mA
		Total of all pins 280 mA	P00 to P03, P30 to P33	30	mA
			P10 to P17, P40 to P47, P64 to P67, P70 to P73	50	mA
			TW0TO0 to TW0TO5	100	mA
			P50 to P57	100	mA
Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
		In flash memory programming mode		–10 to +85	
Storage temperature	T_{stg}			–40 to +125	°C

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator		Oscillation frequency (f_{XP})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	5.0		20	MHz
Crystal resonator		Oscillation frequency (f_{XP})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	5.0		20	MHz
External clock		X1 input frequency (f_{XP})[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	5.0		20	MHz
		X1 input high-/low-level width (t_{XPH}, t_{XPL})	4.0 V ≤ V_{DD} ≤ 5.5 V	24		100	ns

Note Indicates only oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. Since the CPU is started by the internal oscillation clock after reset is released, check the oscillation stabilization time of the X1 input clock using the oscillation stabilization time status register (OSTC). Determine the oscillation stabilization time of the OSTC register and oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Internal Oscillator Characteristics

(T_A = –40 to +85°C, 3.3 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 3.3 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Internal oscillator	Oscillation frequency (f_R)		120	240	480	kHz

DC Characteristics (1/2)

(T_A = −40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

	Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
	Output current, high	I_{OH}	Per pin	4.0 V ≤ V_{DD} ≤ 5.5 V			–5	mA
			Total of P00 to P03, P30 to P33, P50 to P57	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
			Total of P10 to P17, P40 to P47, P64 to P67, P70 to P73, TW0TO0 to TW0TO5	4.0 V ≤ V_{DD} ≤ 5.5 V			–25	mA
	Output current, low	I_{OL}	Per pin for P00 to P03, P10 to P17, P30 to P33, P40 to P47, P64 to P67, P70 to P73	4.0 V ≤ V_{DD} ≤ 5.5 V			10	mA
			Per pin for P50 to P57, TW0TO0 to TW0TO5	4.0 V ≤ V_{DD} ≤ 5.5 V			15	mA
			Total of P00 to P03, P30 to P33	4.0 V ≤ V_{DD} ≤ 5.5 V			15	mA
			Total of P10 to P17, P40 to P47, P64 to P67, P70 to P73	4.0 V ≤ V_{DD} ≤ 5.5 V			25	mA
			Total of TW0TO0 to TW0TO5	4.0 V ≤ V_{DD} ≤ 5.5 V			70	mA
			Total of P50 to P57	4.0 V ≤ V_{DD} ≤ 5.5 V			70	mA
	Input voltage, high	V_{IH1}	P14, P17, P30 to P33, P40 to P47, P64 to P67, P70 to P73	4.0 V ≤ V_{DD} ≤ 5.5 V	0.7V_{DD}		V_{DD}	V
		V_{IH2}	P00 to P03, P10-P13, P15, P16, P50 to P57, $\overline{RESET}$	4.0 V ≤ V_{DD} ≤ 5.5 V	0.8V_{DD}		V_{DD}	V
		V_{IH3}	P20 to P27[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	0.7AV_{REF}		AV_{REF}	V
		V_{IH4}	X1, X2	4.0 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.5		V_{DD}	V
	Input voltage, low	V_{IL1}	P14, P17, P30 to P33, P40 to P47, P64 to P67, P70 to P73	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.3V_{DD}	V
		V_{IL2}	P00 to P03, P10 to P13, P15, P16, P50 to P57, $\overline{RESET}$	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.2V_{DD}	V
		V_{IL3}	P20 to P27[Note]	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.3AV_{REF}	V
		V_{IL4}	X1, X2	4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.4	V
<R>	Output voltage, high	V_{OH}	P00 to P03, P10 to P17, P30 to P33, P40 to P47, P64 to P67, P70 to P73	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –5 mA	V_{DD} – 1.0			V
			P50 to P57, TW0TO0 to TW0TO5	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = –1 mA	V_{DD} – 1.0			V
			I_{OH} = –100 μA	4.0 V ≤ V_{DD} ≤ 5.5 V	V_{DD} – 0.5			V
	Output voltage, low	V_{OL}	P50 to P57, TW0TO0 to TW0TO5	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 15 mA		0.4	2.0	V
<R> <R>			P00 to P03, P10 to P17, P30 to P33, P40 to P47, P64 to P67, P70 to P73 Total I_{OL} = 20 mA	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 10 mA			1.5	V
			I_{OL} = 400 μA	4.0 V ≤ V_{DD} ≤ 5.5 V			0.5	V

Note When used as digital input ports, set AV_{REF} = EV_{DD}.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (2/2)
(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	$V_I = V_{DD}$	P00 to P03, P10 to P17, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P73, $\overline{\text{RESET}}$				3	μA
		$V_I = AV_{REF}$	P20 to P27				3	μA
	I_{LIH2}	$V_I = V_{DD}$	X1, X2[Note 1]				20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	P00 to P03, P10 to P17, P20 to P27, P30 to P33, P40 to P47, P50 to P57, P64 to P67, P70 to P73, $\overline{\text{RESET}}$				–3	μA
	I_{LIL2}		X1, X2[Note 1]				–20	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD}$					3	μA
Output leakage current, low	I_{LOL}	V_O = 0 V					–3	μA
Pull-up resistance value	R_L	V_I = 0 V			10	30	100	kΩ
FLMD0 supply voltage	Flmd	In normal operation mode			0		0.2V_{DD}	V
<R> Supply current[Note 2]	I_{DD1}	X1 crystal oscillation operating[Note 3]	f_{XP} = 20 MHz V_{DD} = 5.0 V ±10%[Note 4]	When A/D converter is stopped		18	36	mA
				When A/D converter is operating[Note 5]		20	40	mA
	I_{DD2}	X1 crystal oscillation HALT mode	f_{XP} = 20 MHz V_{DD} = 5.0 V ±10%	When peripheral functions are stopped		3.5	7.0	mA
				When peripheral functions are operating			15	mA
	I_{DD3}	Internal oscillation clock operating mode[Note 6]	V_{DD} = 5.0 V ±10%			3.0	6.0	mA
	I_{DD4}	STOP mode	V_{DD} = 5.0 V ±10%	Internal oscillator: OFF		3.5	35.5	μA
				Internal oscillator: ON		17.5	63.5	μA

Notes 1. When the inverse level of X1 is input to X2.

2. Total current flowing through the internal power supply (V_{DD}). Peripheral operation current is included (however, the current that flows through the pull-up resistors of ports is not included).

3. I_{DD1} includes peripheral operation current.

4. When PCC = 00H.

5. Including the current that flows through the AV_{REF} pin.

6. When X1 oscillation is stopped.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

AC Characteristics

(1) Basic operation

(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

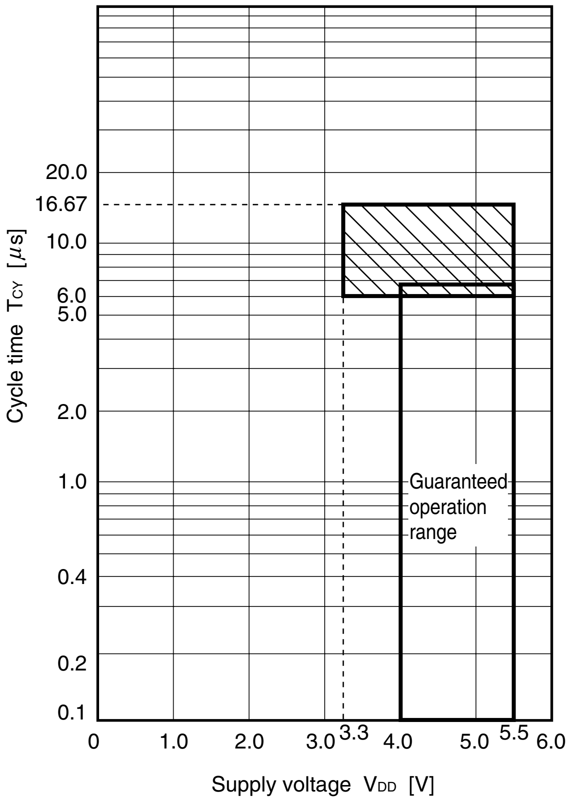
	Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
<R>	Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock operation	X1 input clock		0.1		6.4	μs
				Internal oscillation clock	3.3 V ≤ V_{DD} ≤ 5.5 V	4.17	8.33	16.67	μs
	TI000, TI001 input high-level width, low-level width	t_{TIH0}, t_{TIL0}	4.0 V ≤ V_{DD} ≤ 5.5 V			2/f_{sam1} + 0.1 Note 1			μs
	TI50, TI51 input frequency	f_{TI5}	4.0 V ≤ V_{DD} ≤ 5.5 V					10	MHz
	TI50, TI51 input high-level width, low-level width	t_{TIH5}, t_{TIL5}	4.0 V ≤ V_{DD} ≤ 5.5 V			50			ns
	TIT20IUD, TIT20CUD, TIT20CC0, TIT20CC1, TIT20CLR input high-level width, low-level width	t_{WUDH}, t_{WUDL}	4.0 V ≤ V_{DD} ≤ 5.5 V			2/f_{sam2}+ 0.1 Note 2			μs
	Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	4.0 V ≤ V_{DD} ≤ 5.5 V			1			μs
<R>	ADTRG input high-level width, low-level width	t_{ADTIH}, t_{ADTIL}	4.0 V ≤ V_{DD} ≤ 5.5 V			1			μs
	$\overline{\text{RESET}}$ low-level width	t_{RSL}	4.0 V ≤ V_{DD} ≤ 5.5 V			10			μs

Notes 1. Selection of f_{sam1} = f_{XP}/2, f_{XP}/4, or f_{XP}/256 is possible using bits 0 and 1 (PRM000, PRM001) of prescaler mode register 00 (PRM00). Note that when selecting the TI000 valid edge as the count clock, f_{sam1} = f_{XP}.

2. Selection of f_{sam2} = $f_X/2^3$, $f_X/2^2$, f_X/2, or f_X is possible using bits 0 and 1 (NRC10, NRC11) of noise elimination time select register 1 (NCR1).

<R>

T_{CY} vs. V_{DD} (Main System Clock Operation)



Remark The values indicated by the shaded section are only when the internal oscillation clock is selected.

(2) Serial interface

(T_A = –40 to +85°C, 4.0≤ V_{DD} = EV_{DD} ≤ 5.5 V, 4.0≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(a) UART mode (UART0, dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					312.5	kbps

(b) 3-wire serial I/O mode (master mode, $\overline{\text{SCK10}}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} ≤ 5.5 V	200			ns
$\overline{\text{SCK10}}$ high-/low-level width	t_{KH1}, t_{KL1}		$t_{KCY1}/2 - 10$			ns
SI10 setup time (to $\overline{\text{SCK10}}$↑)	t_{SIK1}		30			ns
SI10 hold time (from $\overline{\text{SCK10}}$↑)	t_{KSI1}		30			ns
Delay time from $\overline{\text{SCK10}}$↓ to SO10 output	t_{KSO1}	C = 100 pF[Note]			30	ns

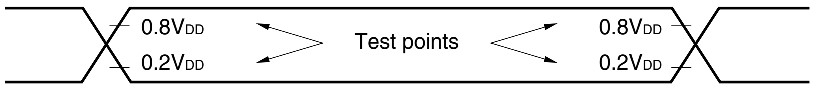
Note C is the load capacitance of the $\overline{\text{SCK10}}$ and SO10 output lines.

(c) 3-wire serial I/O mode (slave mode, $\overline{\text{SCK10}}$... external clock input)

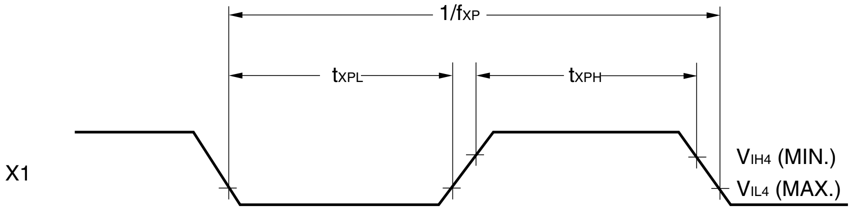
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{\text{SCK10}}$ cycle time	t_{KCY2}		400			ns
$\overline{\text{SCK10}}$ high-/low-level width	t_{KH2}, t_{KL2}		$t_{KCY2}/2$			ns
SI10 setup time (to $\overline{\text{SCK10}}$↑)	t_{SIK2}		80			ns
SI10 hold time (from $\overline{\text{SCK10}}$↑)	t_{KSI2}		50			ns
Delay time from $\overline{\text{SCK10}}$↓ to SO10 output	t_{KSO2}	C = 100 pF[Note]			120	ns

Note C is the load capacitance of the SO10 output line.

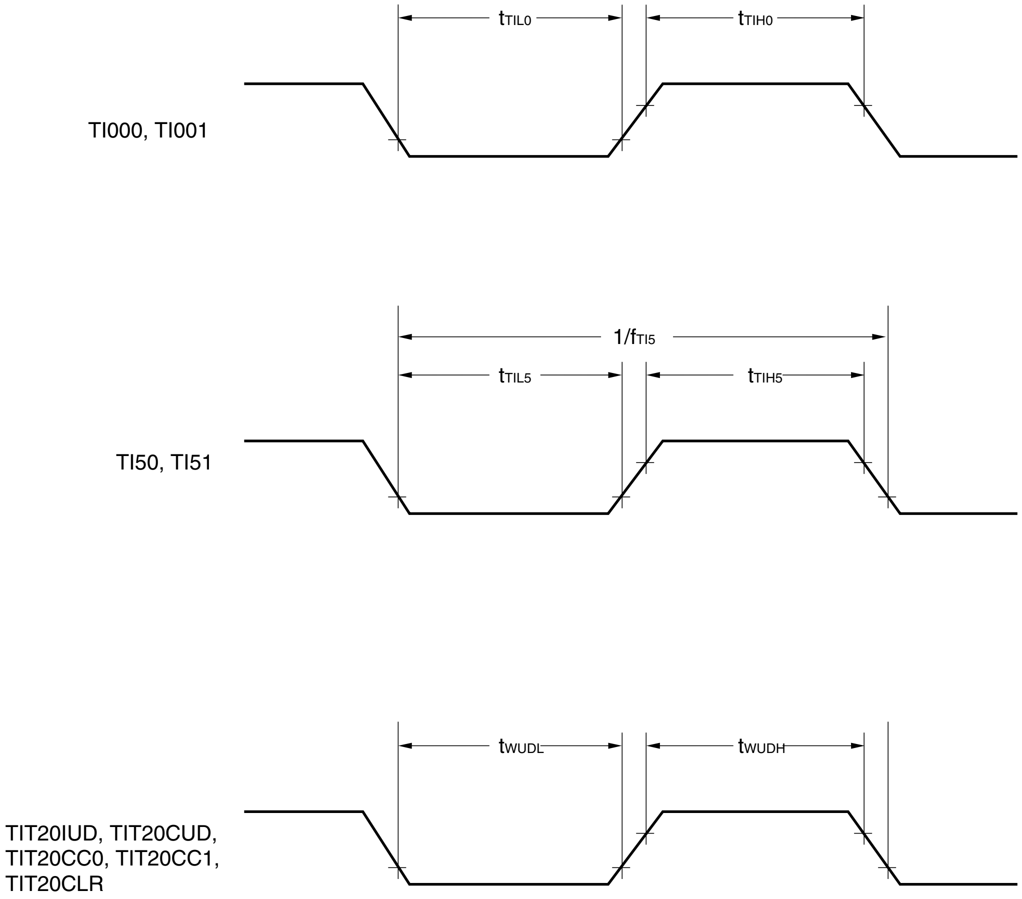
AC Timing Test Points (Excluding X1 Input)



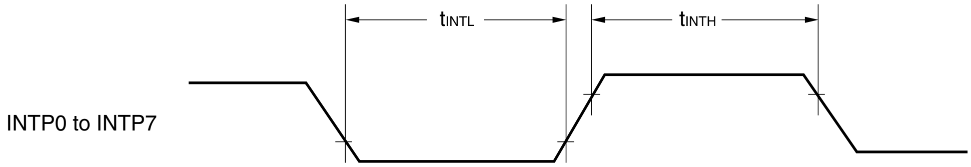
Clock Timing



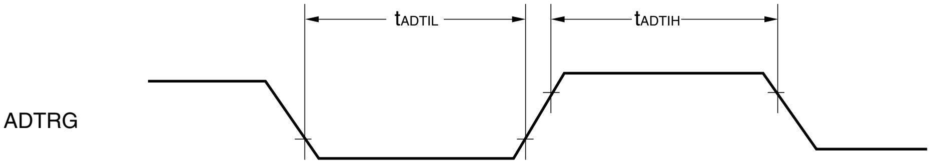
TI Timing



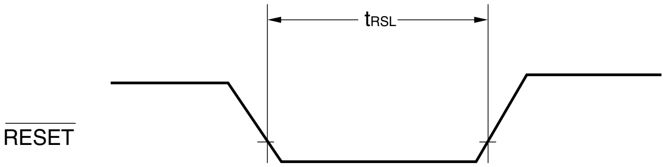
Interrupt Request Input Timing



<R> **A/D Trigger Input Timing**

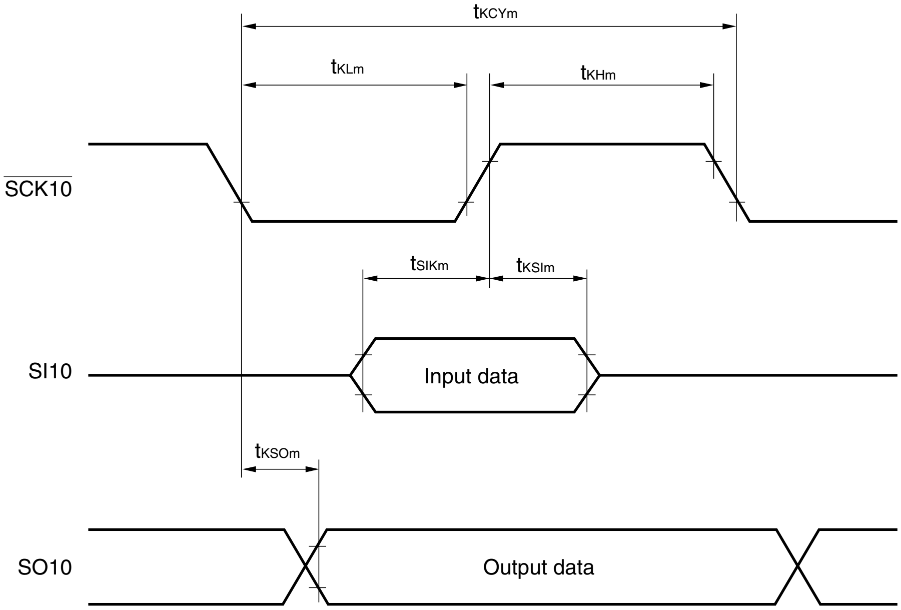


$\overline{\text{RESET}}$ Input Timing



Serial Transfer Timing

3-wire serial I/O mode:



Remark m = 1, 2

A/D Converter Characteristics
(T_A = –40 to +85°C, 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V		±0.2	±0.4	%FSR
<R> Conversion time	t_{CONV}	4.5 V ≤ AV_{REF} ≤ V_{DD}	3.6		100	μs
		4.0 V ≤ AV_{REF} ≤ V_{DD}	4.8		100	μs
Zero-scale error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
Full-scale error[Notes 1, 2]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±0.4	%FSR
Integral non-linearity error[Note 1]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±2.5	LSB
Differential non-linearity error[Note 1]		4.0 V ≤ AV_{REF} ≤ 5.5 V			±1.5	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF}	V

Notes 1. Excludes quantization error (±1/2 LSB).

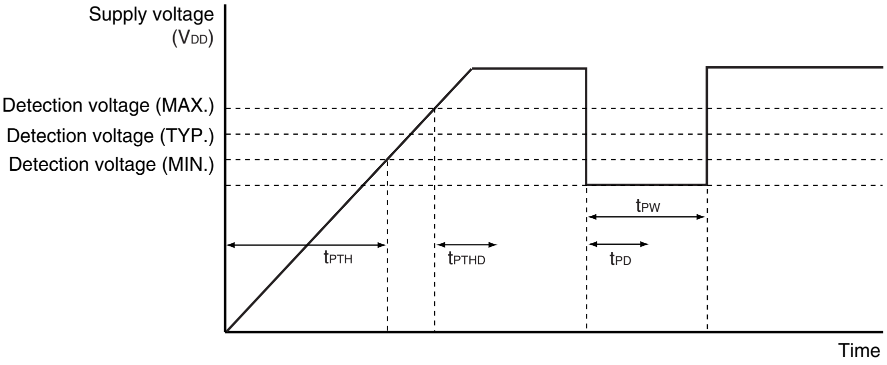
2. This value is indicated as a ratio (%FSR) to the full-scale value.

POC Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC}		3.3	3.5	3.7	V
Power supply rise time	t_{PTH}	V_{DD}: 0 V → 3.3 V	0.002			ms
Response delay time 1[Note]	t_{PTHD}	When power supply rises, after reaching detection voltage (MAX.)			3.0	ms
Response delay time 2[Note]	t_{PD}	When V_{DD} falls			1.0	ms
Minimum pulse width	t_{PW}		0.2			ms

Note Time required from voltage detection to reset release.

POC Circuit Timing



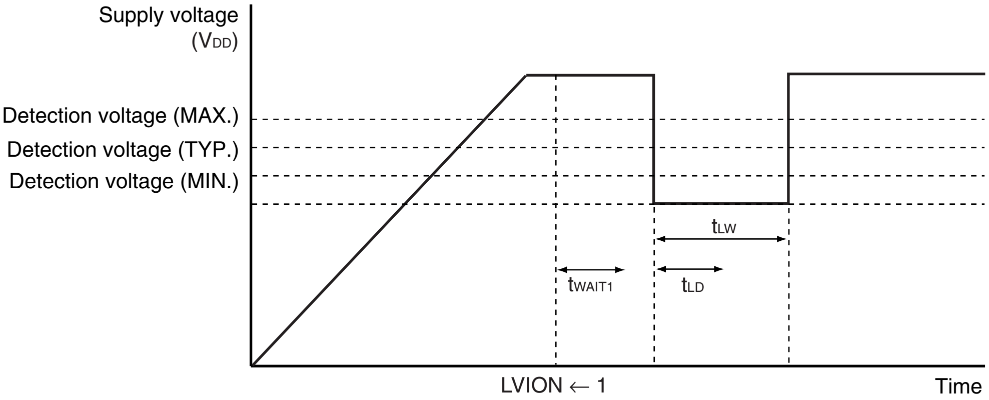
LVI Circuit Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI}		4.1	4.3	4.5	V
Response time[Note 1]	t_{LD}			0.2	2.0	ms
Minimum pulse width	t_{LW}		0.2			ms
Operation stabilization wait time[Note 2]	t_{LWAIT1}			0.1	0.2	ms

Notes 1. Time required from voltage detection to interrupt output or internal reset output.

2. Time required from setting LVION to 1 to operation stabilization.

LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Data retention supply voltage	V_{DDDR}		2.0		5.5	V
Release signal set time	t_{SREL}		0			μs

Flash Memory Programming Characteristics

(T_A = +10 to +85°C, 4.0 V ≤ V_{DD} ≤ 5.5 V, 4.0 V ≤ AV_{REF} ≤ V_{DD}, V_{SS} = 0 V)

<R>

(1) Basic characteristics

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
V_{DD} supply current		I_{DD}	f_X = 20 MHz, V_{DD} = 5.5 V			42	mA
Step erase time		T_{erass}			10		ms
Erase time[Note 1]	Chip unit	T_{eraca}			0.01	2.55	s
	Sector unit	T_{erasa}			0.01	2.55	s
Write time		T_{wrwa}			50	500	μs
Number of rewrites per chip		C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note 2]			100	Times

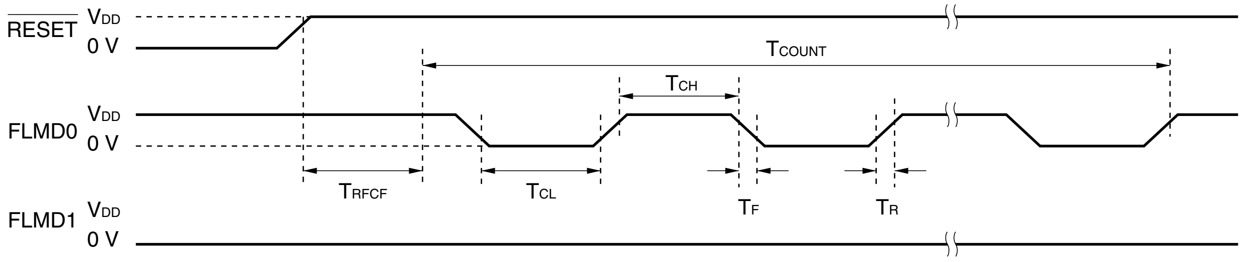
Notes 1. The erase verify time (writeback time) is not included.

2. If a sector (2 KB) is erased after it was written in 512 operations, in word units, the number of rewrite operations is 1. Writing to the same address more than once for one erase is prohibited.

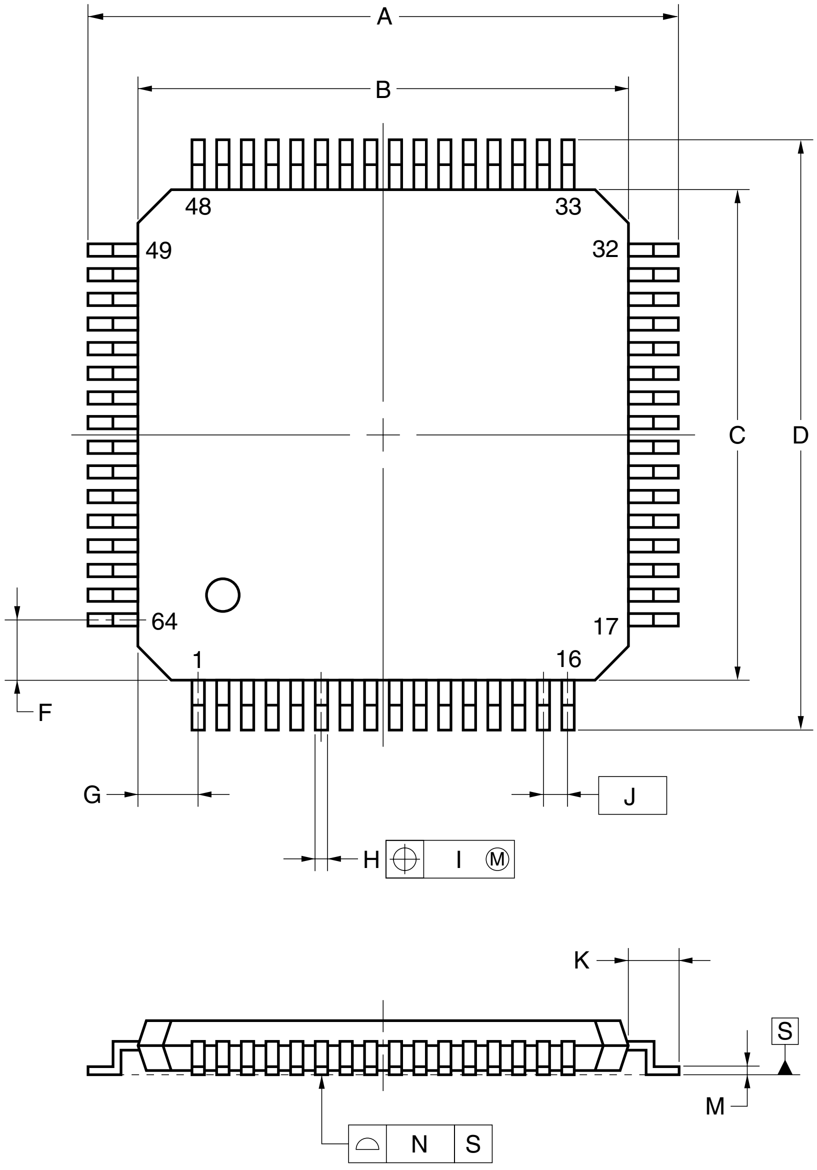
(2) Serial write operation characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Time from $\overline{\text{RESET}}$↑ to FLMD0 count start	T_{RFCF}		$2^{19}/f_X + \alpha$			μs
Count execution time	T_{COUNT}				10	ms
FLMD0 counter high-/low-level width	T_{CH}/T_{CL}		10		100	μs

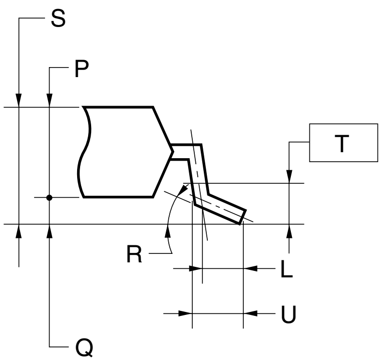
Serial Write Operation



64-PIN PLASTIC TQFP (12x12)



detail of lead end



ITEM	MILLIMETERS
A	14.0±0.2
B	12.0±0.2
C	12.0±0.2
D	14.0±0.2
F	1.125
G	1.125
H	$0.32^{+0.06}_{-0.10}$
I	0.13
J	0.65 (T.P.)
K	1.0±0.2
L	0.5
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	1.0
Q	0.1±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.1±0.1
T	0.25
U	0.6±0.15

P64GK-65-9ET-3

NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

CHAPTER 30 CAUTIONS FOR WAIT

30.1 Cautions for Wait

This product has two internal system buses.

One is a CPU bus and the other is a peripheral bus that interfaces with the low-speed peripheral hardware.

Because the clock of the CPU bus and the clock of the peripheral bus are asynchronous, unexpected illegal data may be passed if an access to the CPU conflicts with an access to the peripheral hardware.

When accessing the peripheral hardware that may cause a conflict, therefore, the CPU repeatedly executes processing, until the correct data is passed.

As a result, the CPU does not start the next instruction processing but waits. If this happens, the number of execution clocks of an instruction increases by the number of wait clocks (for the number of wait clocks, see **Table 30-1**). This must be noted when real-time processing is performed.

30.2 Peripheral Hardware That Generates Wait

Table 30-1 shows the registers that generate a wait request when an access from the CPU is made, and the number of wait clocks of the CPU (when VSWC = 0).

Table 30-1. Registers That Generate Wait and Number of CPU Wait Clocks (When VSWC = 0)

<table>
<tr><th>Peripheral Hardware</th><th>Register</th><th>Access</th><th>Number of Wait Clocks</th></tr>
<tr><td>Watchdog timer</td><td>WDTM</td><td>Write</td><td>3 clocks (fixed)</td></tr>
<tr><td>Serial interface UART00</td><td>ASIS00</td><td>Read</td><td>1 clock (fixed)</td></tr>
<tr><td rowspan="6">A/D converter</td><td>ADM</td><td>Write</td><td rowspan="4">1 to 5 clocks[Note]
(when ADM.5 flag = “1”)
1 to 9 clocks[Note]
(when ADM.5 flag = “0”)</td></tr>
<tr><td>ADS</td><td>Write</td></tr>
<tr><td>PFM</td><td>Write</td></tr>
<tr><td>PFT</td><td>Write</td></tr>
<tr><td>ADCR</td><td>Read</td><td>1 to 5 clocks
(when ADM.5 flag = “1”)
1 to 9 clocks
(when ADM.5 flag = “0”)</td></tr>
<tr><td colspan="3">&lt;Calculating maximum number of wait clocks>
$\{(1/f_{MACRO}) \times 2/(1/f_{CPU})\} + 1$
* The result after the decimal point is truncated if it is less than t_{CPUL} after it has been multiplied by ($1/f_{CPU}$), and is rounded up if it exceeds t_{CPUL}.
f_{MACRO}: Macro operating frequency
(When bit 5 (FR2) of ADM = “1”: $f_X/2$, when bit 5 (FR2) of ADM = “0”: $f_X/2^2$)
f_{CPU}: CPU clock frequency
t_{CPUL}: Low-level width of CPU clock</td></tr>
</table>

Note No wait cycle is generated for the CPU if the number of wait clocks calculated by the above expression is 1.

Remark The clock is the CPU clock (f_{CPU}).

30.3 Example of Wait Occurrence

<1> Watchdog timer

<On execution of MOV WDTM, A>

Number of execution clocks: 8

(5 clocks when data is written to a register that does not issue a wait (MOV sfr, A).)

<On execution of MOV WDTM, #byte>

Number of execution clocks: 10

(7 clocks when data is written to a register that does not issue a wait (MOV sfr, #byte).)

<2> Serial interface UART00

<On execution of MOV A, ASIS00>

Number of execution clocks: 6

(5 clocks when data is read from a register that does not issue a wait (MOV A, sfr).)

<3> A/D converter

Table 30-2. Number of Wait Clocks and Number of Execution Clocks on Occurrence of Wait (A/D Converter)

<On execution of MOV ADM, A; MOV ADS, A; or MOV A, ADCR>

- When f_X = 10 MHz, t_{CPUL} = 50 ns

Value of Bit 5 (FR2) of ADM Register	f_{CPU}	Number of Wait Clocks	Number of Execution Clocks
0	f_X	9 clocks	14 clocks
	$f_X/2$	5 clocks	10 clocks
	$f_X/2^2$	3 clocks	8 clocks
	$f_X/2^3$	2 clocks	7 clocks
	$f_X/2^4$	0 clocks (1 clock^Note^)	5 clocks (6 clocks^Note^)
1	f_X	5 clocks	10 clocks
	$f_X/2$	3 clocks	8 clocks
	$f_X/2^2$	2 clocks	7 clocks
	$f_X/2^3$	0 clocks (1 clock^Note^)	5 clocks (6 clocks^Note^)
	$f_X/2^4$	0 clocks (1 clock^Note^)	5 clocks (6 clocks^Note^)

Note On execution of MOV A, ADCR

Remark The clock is the CPU clock (f_{CPU}).

f_X: X1 input clock frequency

t_{CPUL}: Low-level width of CPU clock

<R>

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the μ PD78F0714.
Figure A-1 shows the development tool configuration.

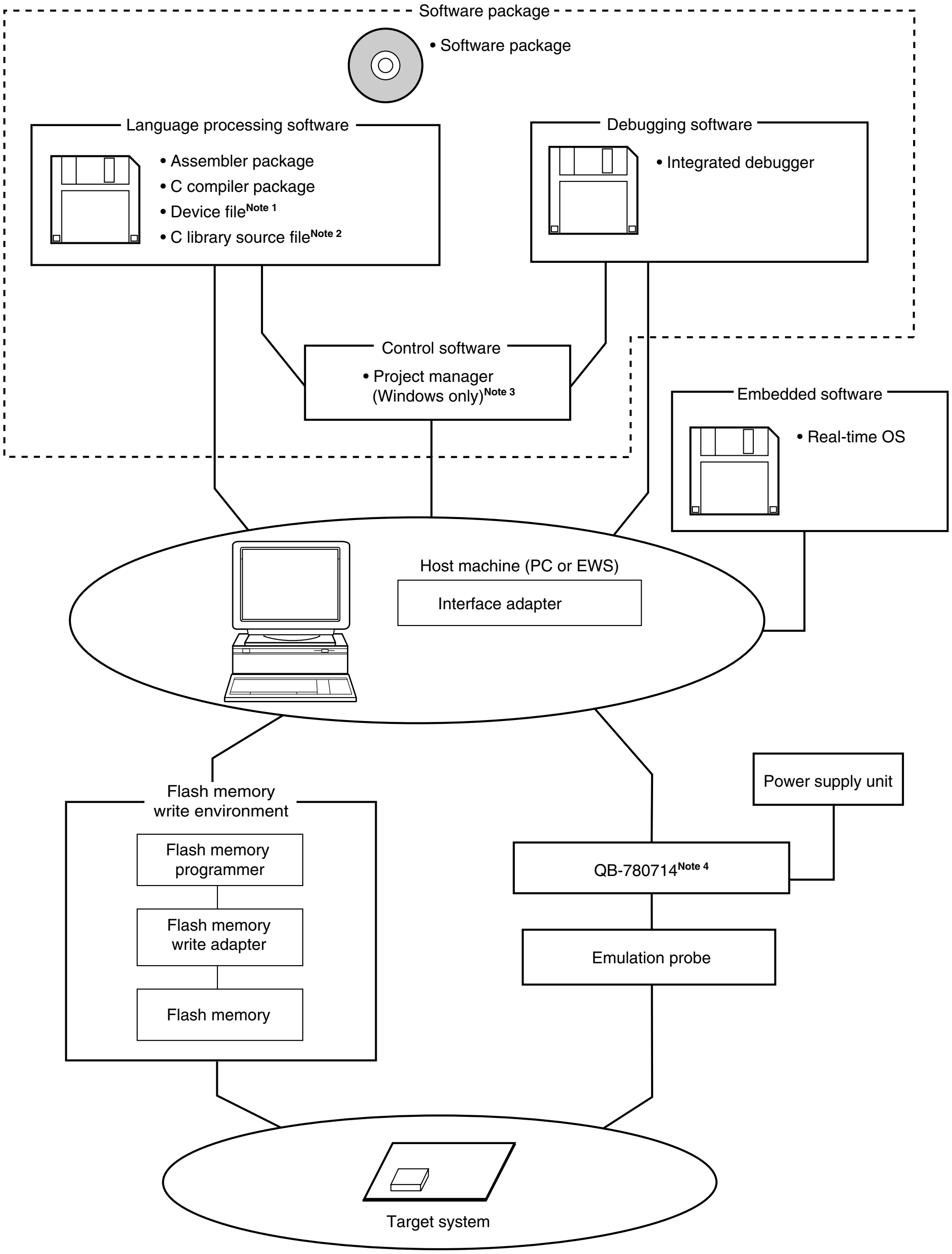
- **Support for PC98-NX series**
  Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows**
  Unless otherwise specified, "Windows" means the following OSs.
  - Windows 3.1
  - Windows 95, Windows 98, Windows 2000, Windows XP
  - Windows NT™ Ver 4.0

Caution For the development tools of the μ PD78F0714, contact an NEC Electronics sales representative.

Figure A-1. Development Tool Configuration (1/3)

(1) When using the in-circuit emulator QB-780714



Notes 1. Download the device file for μPD78F0714 (DF780714) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

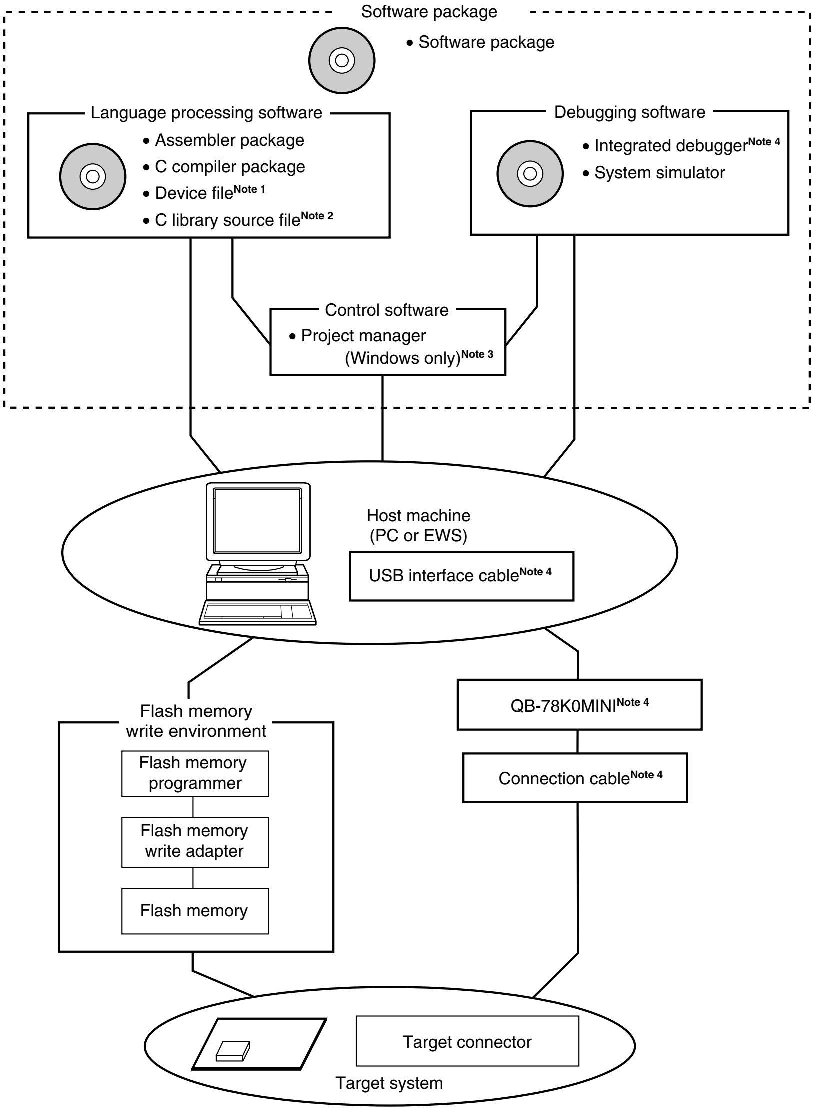
2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

4. In-circuit emulator QB-780714 is supplied with integrated debugger ID78K0-QB, flash memory programmer PG-FPL (or QB-MINI2), power supply unit, and USB interface cable. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

Figure A-1. Development Tool Configuration (2/3)

(2) When using the on-chip debug emulator QB-78K0MINI



Notes 1. Download the device file for μPD78F0714 (DF780714) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

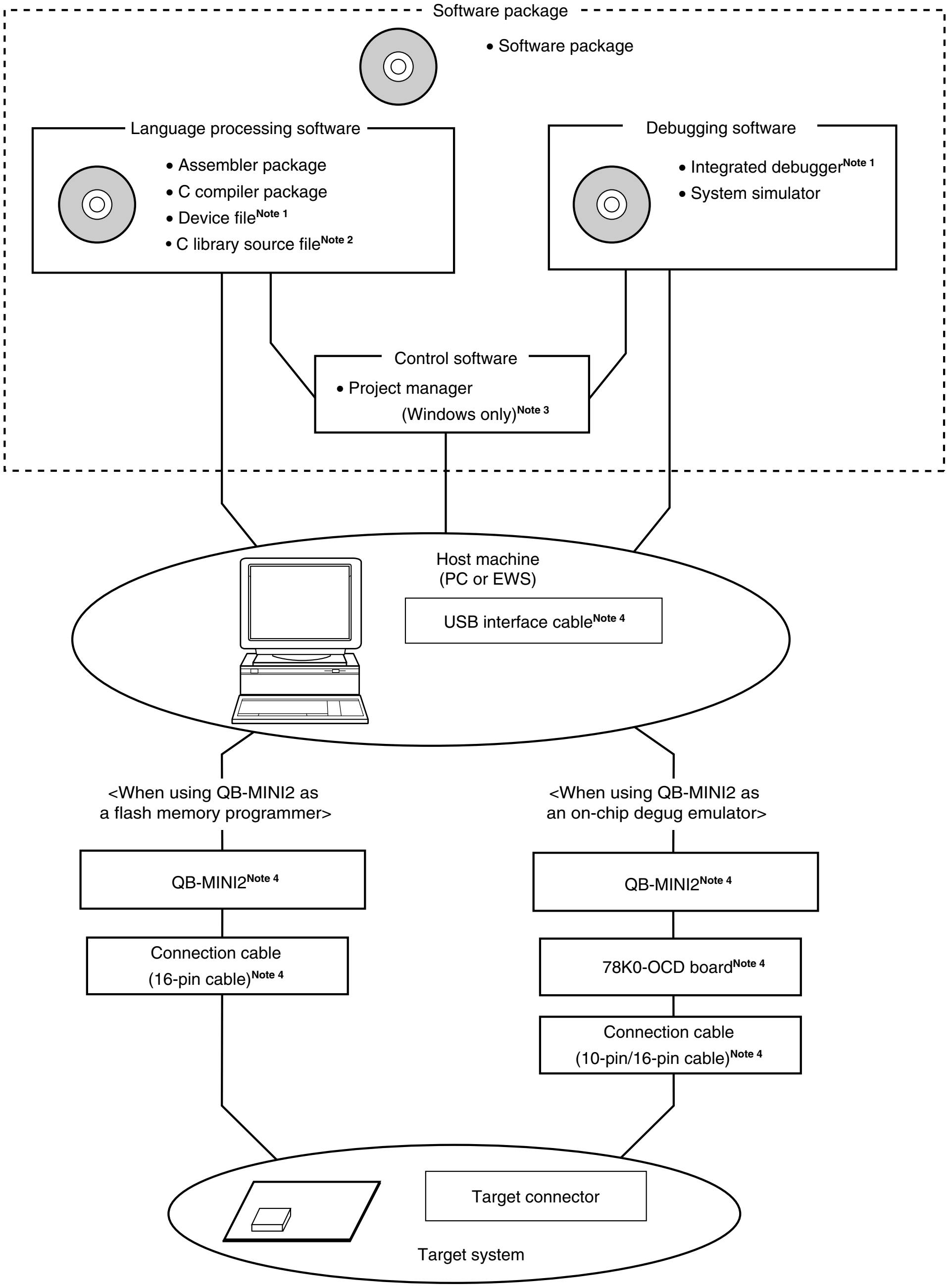
2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

4. QB-78K0MINI is supplied with integrated debugger ID78K0-QB, USB interface cable, and connection cable. Any other products are sold separately.

Figure A-1. Development Tool Configuration (3/3)

(3) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for μPD78F0714 (DF780714) and the integrated debugger (ID78K0-QB) from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The C library source file is not included in the software package.

3. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

4. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. Any other products are sold separately. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.1 Software Package

SP78K0 78K/0 Series software package	Development tools (software) common to the 78K/0 Series are combined in this package.
	Part number: µS××××SP78K0

Remark ×××× in the part number differs depending on the host machine and OS used.

µS××××SP78K0

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

RA78K0 Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF780714) (sold separately). **<Precaution when using RA78K0 in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: µS××××RA78K0
CC78K0 C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file (both sold separately). **<Precaution when using CC78K0 in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
	Part number: µS××××CC78K0
DF780714[Note 1] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0, CC78K0, and ID78K0-QB) (all sold separately). The corresponding OS and host machine differ depending on the tool to be used.
	Part number: µS××××DF780714
CC78K0-L[Note 2] C library source file	This is a source file of the functions that configure the object library included in the C compiler package. This file is required to match the object library included in the C compiler package to the user's specifications.
	Part number: µS××××CC78K0-L

Notes 1. The DF780714 can be used in common with the RA78K0, CC78K0, and ID78K0-QB. Download the DF780714 from the download site for development tools (http://www.necel.com/micro/ods/eng/index.html).

2. The CC78K0-L is not included in the software package (SP78K0).

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××RA78K0
μS××××CC78K0
μS××××CC78K0-L

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3P17	HP9000 series 700™	HP-UX™ (Rel. 10.10)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4) Solaris™ (Rel. 2.5.1)	

μS××××DF780714

××××	Host Machine	OS	Supply Medium
AB13	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	3.5-inch 2HD FD
BB13		Windows (English version)	

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from the project manager. **<Caution>** The project manager is included in the assembler package (RA78K0). It can only be used in Windows.

A.4 Flash Memory Programming Tools

A.4.1 When using flash memory programmer PG-FP5, FL-PR5, PG-FP4, FL-PR4, and PG-FPL

FL-PR4, PG-FP4, PL-PR5, PG-FP5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
PG-FPL Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory. Included with in-circuit emulator QB-780714.
FA-64GK-9ET-A Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use. • FA-64GK-9ET-A: For 64-pin plastic TQFP (GK-9ET type)

Remarks 1. FL-PR5, FL-PR4, and FA-64GK-9ET-A are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172 Naito Densei Machida Mfg. Co., Ltd.

2. Use the latest version of the flash memory programming adapter.

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the μPD78F0714. When using this as flash memory programmer, it should be used in combination with a connection cable (16-pin cable) and a USB interface cable that is used to connect the host machine.
Target connector specifications	16-pin general-purpose connector (2.54 mm pitch)

Remarks 1. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. A connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.

2. Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.5 Debugging Tools (Hardware)

A.5.1 When using in-circuit emulator QB-780714

QB-780714[Note] In-circuit emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using the μPD78F0714. It supports the integrated debugger (ID78K0-QB). This emulator should be used in combination with a power supply unit and emulation probe. USB is used to connect this emulator to the host machine.
QB-144-CA-01 Check pin adapter	This adapter is used in waveform monitoring using the oscilloscope, etc.
QB-80-EP-01T Emulation probe	This is a flexible type probe used to connect the in-circuit emulator to the target system.
QB-64GK-EA-01T Exchange adapter	This adapter is used to perform the pin conversion from the in-circuit emulator to the target connector. • QB-64GK-EA-01T: For 64-pin plastic TQFP (GK-9ET type)
QB-64GK-YS-01T Space adapter	This adapter is used to adjust the height between the target system and in-circuit emulator if required. • QB-64GK-YS-01T: For 64-pin plastic TQFP (GK-9ET type)
QB-64GK-YQ-01T YQ connector	This connector is used to connect the target connector to the exchange adapter. • QB-64GK-YQ-01T: For 64-pin plastic TQFP (GK-9ET type)
QB-64GK-HQ-01T Mount adapter	This adapter is used to mount the target device onto the target device with socket. • QB-64GK-HQ-01T: For 64-pin plastic TQFP (GK-9ET type) with on-chip debug function
QB-64GK-NQ-01T Target connector	This connector is used to mount the in-circuit emulator onto the target system. • QB-64GK-NQ-01T: For 64-pin plastic TQFP (GK-9ET type)

Note The QB-780714 is supplied with a power supply unit, USB interface cable, and flash memory programmer PG-FPL. It is also supplied with integrated debugger ID78K0-QB as control software.

Remark The package contents differ depending on the part number.

Package Contents Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	YQ Connector	Target Connector
QB-780714-ZZZ	QB-780714	Not included			
QB-780714-T30MC		QB-80-EP-01T	QB-64GK-EA-01T	QB-64GK-YQ-01T	QB-64GK-NQ-01T

A.5.2 When using on-chip debug emulator QB-78K0MINI

QB-78K0MINI[Note] On-chip debug emulator	The on-chip debug emulator serves to debug hardware and software when developing application systems using the μPD78F0714. It supports the integrated debugger (ID78K0-QB) supplied with the QB-78K0MINI. This emulator uses a connection cable and a USB interface cable that is used to connect the host machine.
Target connector specifications	10-pin general-purpose connector (2.54 mm pitch)

Note The QB-78K0MINI is supplied with a USB interface cable and a connection cable. It is also supplied with integrated debugger ID78K0-QB as control software.

A.5.3 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the μPD78F0714. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. When using this as on-chip debug emulator, it should be used in combination with a connection cable (10-pin cable or 16-pin cable), a USB interface cable that is used to connect the host machine, and the 78K0-OCD board.
Target connector specifications	10-pin general-purpose connector (2.54 mm pitch) or 16-pin general-purpose connector (2.54 mm pitch)

Remarks 1. The QB-MINI2 is supplied with a USB interface cable and connection cables (10-pin cable and 16-pin cable), and the 78K0-OCD board. A connection cable (10-pin cable) and the 78K0-OCD board are used only when using the on-chip debug function.

2. Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www.necel.com/micro/en/development/asia/minicube2/minicube2.html).

A.6 Debugging Tools (Software)

ID78K0-QB Integrated debugger	This debugger supports the in-circuit emulators for the 78K/0 microcontrollers. The ID78K0-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (sold separately).
	Part number: μS××××ID78K0-QB

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID78K0-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

APPENDIX B REGISTER INDEX

B.1 Register Index (In Alphabetical Order with Respect to Register Names)

B.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

<R>

APPENDIX C REVISION HISTORY

C.1 Major Revisions in This Edition

(1/3)

Page	Description	Classification
INTRODUCTION		
p.7	Change of **Documents Related to Development Tools (Software) (User's Manuals)**	(e)
p.7	Change of **Documents Related to Development Tools (Hardware) (User's Manuals)**	(e)
CHAPTER 1 OUTLINE		
p.17	Change of **1.2 Applications**	(c)
p.21	Change of "Vectored interrupt sources" in **1.6 Outline of Functions**	(d)
CHAPTER 2 PIN FUNCTIONS		
p.28	Addition of **Caution** to **2.2.3 (1) Port mode**	(c)
CHAPTER 3 CPU ARCHITECTURE		
p.51	Change of **3.3.3 Table indirect addressing**	(c)
CHAPTER 4 PORT FUNCTIONS		
p.70	Addition of **Caution** to **4.2.3 Port 2**	(c)
p.83	Addition of **4.5 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)**	(c)
CHAPTER 6 10-BIT INVERTER CONTROL TIMER		
p.107	Change of **Figure 6-1 Block Diagram of 10-Bit Inverter Control Timer**	(a)
CHAPTER 13 REAL-TIME OUTPUT PORT		
pp.239, 240	Change of **Figure 13-1 Block Diagram of Real-Time Output Port**	(a)
p.241	Deletion of **Notes1** in **Table 13-2 Operation During Manipulation of Real-Time Output Buffer Register 0**	(c)
p.242	Deletion of **Notes1** in **Table 13-3 Operation During Manipulation of Real-Time Output Buffer Register 1**	(c)
p.244	Change of **Figure 13-6 Format of Real-Time Output Port Mode Register 1**	(c)
p.255	Change of **Table 13-7 Relationship Between Settings of Each Bit of Control Register and Real-Time Output**	(a)
CHAPTER 14 DC INVERTER CONTROL FUNCTION		
p.261	Addition of description	(c)
CHAPTER 15 A/D CONVERTER		
p.266	Change of **Figure 15-3 Format of A/D Converter Mode Register (ADM)**	(b)
p.268	Addition of **Table 15-3 A/D Conversion Time**	(b)
CHAPTER 16 SERIAL INTERFACE UART00		
p.288	Addition of **Caution** to **16.1 (2) Asynchronous serial interface (UART) mode**	(c)
p.291	Addition of **Caution** to **16.2 (3) Transmit shift register 00 (TXS00)**	(c)
p.293	Addition of **Caution** to **Figure 16-2 Format of Asynchronous Serial Interface Operation Mode Register 00 (ASIM00)**	(c)
p.298	Change of procedure of setting an operation in **16.4.2 (1) Registers used**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/3)

Page	Description	Classification
CHAPTER 19 INTERRUPT FUNCTIONS		
p.335	Addition of **19.1 (1) Non-maskable interrupt**	(b)
p.336	Change of **Table 19-1 Interrupt Source List (1/2)**	(b)
p.337	Addition of **(A) Internal non-maskable interrupt** to **Figure 19-1 Basic Configuration of Interrupt Function (1/2)**	(b)
p.342	Change of **Figure 19-2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H)**	(b)
p.343	Change of **Figure 19-3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H)**	(b)
p.344	Change of **Figure 19-4 Format of Priority Specification Flag Registers (PR0L, PR0H, PR1L, PR1H)**	(b)
p.347	Addition of **19.4.1 Non-maskable interrupt request acknowledgment operation**	(b)
p.352	Addition of description to **19.4.4 Multiple interrupt servicing**	(b)
p.352	Change of **Table 19-5 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing**	(b)
CHAPTER 20 STANDBY FUNCTION		
p.361	Addition of **(b) Release by non-maskable interrupt request** to **(2) HALT mode release**	(b)
p.363	Change of **Table 20-3 Operation in Response to Interrupt Request in HALT Mode**	(b)
p.366	Addition of **(b) Release by non-maskable interrupt request** to **(2) STOP mode release**	(b)
p.367	Change of **Table 20-5 Operation in Response to Interrupt Request in STOP Mode**	(b)
CHAPTER 23 LOW-VOLTAGE DETECTOR		
p.382	Change of **23.4 (1) When used as reset**	(b)
p.383	Change of **Figure 23-3 Timing of Low-Voltage Detector Internal Reset Signal Generation**	(b)
p.384	Change of **23.4 (2) When used as interrupt**	(b)
p.385	Change of **Figure 23-4 Timing of Low-Voltage Detector Interrupt Signal Generation**	(b)
CHAPTER 24 OPTION BYTES		
p.390	Change of **Figure 24-2 Format of Option Bytes**	(b)
CHAPTER 25 FLASH MEMORY		
p.404	Change of **Table 25-5 Communication Modes** and addition of **Note**	(c)
p.406	Change of **Remark** in **25.7 Flash Memory Programming by Self-Writing**	(e)
CHAPTER 26 ON-CHIP DEBUG FUNCTION		
p.413	Change of description and **Remark**	(e)
CHAPTER 27 INSTRUCTION SET		
p.422	Change of CALLT in **27.2 Operation List**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/3)

Page	Description	Classification
CHAPTER 28 ELECTRICAL SPECIFICATIONS		
p.427	Change of **Caution**	(c)
p.430	Change of **Output voltage, high** and **Output voltage, low** in **DC Characteristics (1/2)**	(b)
p.431	Change of **Supply current** in **DC Characteristics (2/2)**	(b)
p.432	Change of **Instruction cycle** and addition of **ADTRG input high-level width, low-level width** in **AC Characteristics**	(b)
pp.433, 436	Change of **T_{CY} vs. V_{DD} (Main System Clock Operation)** and addition of **A/D Trigger Input Timing** in **AC Characteristics**	(b)
p.437	Change of **Conversion time** in **A/D Converter Characteristics**	(b)
p.438	Change of **Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics**	(b)
p.439	Change of **(1) Basic characteristics** in **Flash Memory Programming Characteristics**	(b)
APPENDIX A DEVELOPMENT TOOLS		
Throughout	Revision of chapter	(c)
APPENDIX C REVISION HISTORY		
p.461	Addition of chapter	(c)

Remark “Classification” in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

For further information, please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/